암호와 망보안

리 과 대 학
외국문도서출판사
주체91

암호와 망보안

원리와 실천

리 과 대 학
외국문도서출판사

차 례

2편. 공개열쇠암호와 하쉬함수

6장. 공개열쇠암호

7장. 수론의 초보

8장. 통보문인증과 하쉬함수

9장. 하쉬 및 Mac알고리듬

10장. 수자서명과 인증규약

3편. 망보안실천

11장. 인증응용

12장. 전자우편보안

13장. IP보안

14장. Web보안

4편. 체계보안

15장. 침입자, 비루스와 웜

머 리 말

비루스, 해커, 도청, 위조 등과 관련하여 오늘의 콤퓨터망들에서 **정보보안**이 문제로 되지 않는 때가 거의 없다. 이 책에서는 두가지 중요한 문제를 전제로 한다. 첫째로, 콤퓨터체계들과 망을 통한 그것들 호상련관이 보다 긴밀해짐으로써 그 체계들을 리용하여 통신되고 기억되는 정보에 대한 개인들과 기관들의 의존성이 증대되였다. 이것은 또한 자료와 자원의 로출을 막고 자료와 통보문의 믿음성을 담보하며 망을 통한 공격으로부터 체계를 보호해야 할 필요성을 높이게 하였다. 둘째로, 암호화와 망보안학문이 심화되면서 망보안을 강화할수 있는 응용프로그람들이 개발되여 이미 실천에 리용되고 있다는것이다.

책의 목적

이 책의 목적은 망보안과 암호학의 리론 및 실천의 두 측면에서 실무적개괄을 주는것이다. 이 책의 첫 두개편에서는 망보안능력과 관련한 기본문제들이 암호학과 그리고 망보안기술에 대한 안내자료와 개괄을 주는 방법으로 취급되였다. 이 책의 뒤편들은 망보안실무 즉 망보안을 보장하는데 구현되여 리용되고 있는 실제적인 응용에 대하여 취급하고 있다. 취급되는 문제로부터 이 책의 전반내용은 여러가지 학문분야를 기초로 삼고 서술되였다. 특히 이 책에서 론의되는 일부 문제들의 의의에 대하여서는 수론과 확률론의 기초지식이 없이는 리해할수 없다. 그럼에도 불구하고 이 책은 독자적으로 내용을 구성하는 방향에서 노력을 기울였다. 이 책에서는 필요한 기초수학지식을 주었을뿐아니라 그 지식을 직관적으로 리해할수 있도록 하였다.

이러한 기초지식은 필요한 대목에 주었다. 이런 방법은 책에 포함시킨 자료들이 생동하게 리용되게 하며 책의 시작부터 모든 수학지식을 통털어서 간단히 주는데 더 좋을것이라고 본다.

독자대상

이 책은 과학자들과 전문가들을 대상으로 한다. 교재로서 이 책은 대부분의 콤퓨터과학, 콤퓨터공학, 전자공학을 전공하는 사람들의 암호학과 망보안에 대한 한학기과정안을 취급하였다. 이 책은 또한 기본참고서로도 되며 자습용으로도 리용할수 있다.

책의 구성

이 책의 4개 편을 간단히 소개하면 다음과 같다.

1. **전통암호화:** 기밀성을 위한 전통암호의 리용을 비롯하여 전통암호알고리듬의 구체적인 연구와 설계원리
2. **공개열쇠암호화와 하쉬함수:** 공개열쇠암호알고리듬의 구체적인 연구와 설계원리. 이 편에서는 통보문인증부호, 하쉬함수의 리용과 수자서명, 공개열쇠확인에 대하여 서술하였다.
3. **망보안실천:** Kerberos, X.509v3확인, PGP, S/MIME, IP Security, SL/TLS, SET 등과 같은 중요한 망보안도구들과 응용실례들을 취급하였다.
4. **체계보안:** 신용체계와 방화벽의 리용, 비루스와 침입자에 의한 위협과 그에 대한

대책을 비롯하여 체계준위에서 보안을 취급하였다.

더 구체적으로 1장의 마감에 매장에 대한 개괄 그리고 색인과 자주 리용되는 략어, 참고문헌을 주었다. 매장의 마지막에 문제들과 참고문헌을 주었다.

교원들과 대학생들을 위한 인터네트봉사

대학생들과 교원들이 참고할수 있는 이 책과 관련한 Web페지가 있다. Web페지는 PDF(Adobe Acrobat)형식에서 이 책에 관계되는 Web싸이트에 대한 주소를 준다. 또한 책에 관계되는 인터네트우편목록에 대한 서명정보가 들어 있다. Web페지는 http://www. shore. net/~ws/Security2e. html.이다. 인터네트전자우편목록은 이 책을 리용하는데서 의문, 의견, 정보를 교환할수 있게끔 설정되였다. 인쇄오유나 다른 오유를 발견하면 이 책에 대한 고침표목록을 주소 http:/www. shore. net/~ws에서 리용할수 있다.

암호학과 망보안교수를 위한 실습과제

교원들에게 있어서 암호학이나 보안에 대한 학과목에서 중요한 구성부분은 학생들이 손 쉽게 교재의 개념들을 더 공고히 하게 하는 실습과제 혹은 실습과제들의 모임이다. 이 책은 교수에 실습과제를 결합하도록 상당한 정도의 방조를 제공한다. 교원의 지도서에는 실습과제를 구조화하고 할당하는 문제뿐아니라 교재들에서 주고 있는 광범한 범위의 문제들을 담을수 있게 제안된 실습과제들을 포함한다.

- **탐색실습과제**: 학생들이 인터네트에서 특정의 주제를 찾아서 보고서를 쓸수 있게 가르치는 탐색과제들의 모임
- **프로그람작성실습과제**: 임의의 가동환경에서 어떤 적당한 언어로 실현할수 있는 광범한 범위의 화제거리를 반영하는 프로그람작성실습과제들의 모임
- **읽기/보고서제출과제**: 매장에서 문헌해제로 주는 론문의 목록으로서 학생들이 읽고 간단한 보고서를 작성하도록 과제를 줄수 있다.

이 책에서 새로운것은 무엇인가

이 책의 첫판이 나온 다음에도 이 분야에서는 발명과 개선이 계속되였다. 이 책에서는 전반부분을 폭 넓게 종합적으로 취급하면서도 이러한 내용들도 포함되도록 노력하였다. 이러한 목적에서 첫판을 개정하면서 이 과목을 가르치는 전문가들에게 광범한 심사를 의뢰하였다. 그리하여 많은 문제들이 명백해 지고 째이게 되였으며 삽화의 내용들도 개선되였다. 그리고 새로운 분야의 문제들이 첨부되였다.

이 책의 제목을 암호와 망보안이라고 하였는데 그것은 망보안에서 암호알고리듬이 중심적역할을 한다는것을 반영한다. 또한 이 책은 자습용과 교재로 다 사용할수 있도록 서술에서 론리성을 더 잘 보장하게끔 재편집하였다. 또한 교육학적 및 독자의 편리를 도모하기 위하여 책전반에서 뚜렷한 변화를 주었다. 새로 포함된 중요한 내용은 다음과 같다.

- **블로크암호설계의 새로운 론의**: 블로크암호구조를 론의하는 3개의 절들에 블로크암호설계원리와 최근 개발된 암호들의 특징을 첨부함으로써 전통암호법에 대한 전면적고찰이 더 심화되였다.

- **추가적인 전통암호화알고리듬의 취급:** Blowfish, RC5, CAST-128과 같은 최근 판매되는 제품과 인터네트표준의 암호화알고리듬을 취급한다.
- **타원곡선암호에 대한 새로운 취급:** 이것은 공개열쇠암호에서 RSA와 디피-헬만(Diffie-Hellman)에 대한 중요한 대안이다.
- **수론에 대한 보다 확장된 취급:** 개론이 옹근 한개 장으로까지 확장되였으며 이 문제들에 대한 리해를 더 명백히 하기 위하여 많은 실례를 주었다.
- **하쉬부호와 MAC설계에 대한 새로운 론의:** 통보문인증부호와 하쉬함수의 보안과 설계원리가 첨가되였다.
- **보충적인 하쉬함수와 MAC알고리듬의 새로운 범위:** RIPEMD-160과 HMAC를 비롯한 MAC판매제품과 인터네트표준제품들에서 찾아 보게 되는 최근의 알고리듬을 개괄하였다.
- **X.509범위와 새로운 X.509v3처리에 대한 확장:** X.509의 공개열쇠확인법 특히 판본 3은 많은 제품들과 인터네트표준들에서 새로 찾아 보게 된다.
- **S/MIME에 대한 새로운 범위:** S/MIME은 상업용보안전자우편의 표준으로 되고 있다.
- **IP보안에 대한 새로운 장:** IP보안은 가상사설망구축과 인터네트상의 말단대말단형보안을 위한 새로운 표준들의 중요한 모임이다. 하나의 옹근 장에 이 중요한 문제를 담고 있다.
- **Web보안에 대한 새로운 장:** Web보안은 망보안의 가장 중요한 분야의 하나로 되였으며 여기서 많은 새로운 문제가 제기되고 있다. 하나의 장에 이 문제를 담고 있다. 여기에는 다음과 같은 2개의 주요한 Web보안표준이 있다.
  - 보안소케트층(SSL)과 전송층보안(TLS): SSL은 사실상 모든 열람기와 봉사기제공에 기초한 Web보안을 위한 표준이다. TLS는 SSL을 대신하기 위한 인터네트표준으로 되고 있다.
  - 보안전자트랜잭션(SET): SET는 Web우에서 보안전자상거래용표준으로 출현하고 있다.
- **방화벽에 대한 새로운 장**: 방화벽은 인터네트에 련결되는 싸이트를 보안하는데 적합한 제품으로 출현하였다.
- **새롭게 확장된 교수지원**: 이전과 같이 이 책의 모든 문제들에 대한 풀이를 포함하는 교원들의 지도서이다. 그외 앞에서 이야기한것처럼 학생들의 실습과제지원을 주고 있다.
- **기타 달라 진 내용**
  - 전통암호알고리듬을 위한 X-통의 설계에서 중요한 함수들을 취급한 새로운 절
  - 많은 장들의 참고문헌에 대한 절에서 해당한 Web싸이트들을 소개하고 있다.
  - 수십개의 새로운 자습문제들이 보충되였다.

1장. 소 개

기관들에서의 정보보안에 대한 요구는 최근 수십년동안에 두가지 중대한 변화를 가져 왔다. 자료처리설비가 광범히 리용되기전에 기관들이 중요시한 정보의 보안은 주로 물리적 및 관리적인 방법(수단)에 의하여 실현되였다. 전자에 대한 실례는 중요한 문서를 보관하기 위하여 관건장치된 서류함의 리용이고 후자의 실례는 사람들을 모집할 때 적용되는 선발절차이다.

콤퓨터가 도입되면서 콤퓨터에 기억된 파일들과 기타 정보들을 보호하기 위한 자동화된 도구들에 대한 요구가 높아 졌다. 이것은 시분할체계와 같은 공유체계의 경우 더욱 심각하였다. 더우기 자료망이나 공중전화상으로 접근할수 있는 체계의 경우 더 첨예하였다.

자료보호와 해커들의 칩입에 대응하여 설계된 도구일반에 대한 총칭이 바로 콤퓨터 보안이다.

보안에 영향을 준 두번째 중요한 변화는 **분산체계**의 도입과 콤퓨터와 말단사용자사이, 콤퓨터와 콤퓨터사이에 자료전송을 위한 망 및 통신시설의 리용이다.

전송중에 있는 자료보안에 **망보안**대책이 필요하다. 실제적으로 모든 봉사기관, 정부기관, 과학기관들의 자료처리기들이 호상접속되여 하나의 련결된 망을 이루기때문에 망보안이라는 말을 잘못 인식할수도 있다. 그러한 망모임을 흔히 인터네트라고 한다.

보안의 이 두 형식에는 명백한 경계가 없다. 실례로 정보체계들에서 공격의 가장 공개된 형태의 하나는 콤퓨터비루스이다. 비루스는 그것들이 들어 있는 어떤 디스크가 콤퓨터에 설치되면 물리적으로 어떤 체계에 침습한다. 비루스는 인터네트를 통하여 전염될수도 있다. 어느 경우에나 일단 비루스가 어떤 체계를 전염시키면 콤퓨터안의 보안도구들로써 비루스를 검출하고 제거해야 한다.

이 책은 인터네트보안 즉 정보전송이 동반하는 보안침입들의 검출, 예방, 탐지, 제거에 집중한다. 이것은 모든 가능성을 포함하는 넓은 의미의 주제이다. 이 책에서 취급하고 있는 령역을 독자들에게 알려 줄 목적으로 보안침입에 대한 다음의 실례를 고찰한다.

1. 사용자 A가 사용자 B에게 어떤 파일을 전송한다. 그 파일은 로출되여서는 안될 중요한 정보(실례로 로임명세)를 담고 있다. 그 파일을 읽을 권리가 없는 사용자 C가 자료의 전송과정을 감시하고 있다가 전송중에 그 파일의 복사물을 얻을 수 있다.
2. 망관리자 D는 자기가 관리하는 어떤 콤퓨터 E에 통보를 보낸다. 통보문은 그 콤퓨터에 대한 접근이 허락된 몇명의 사용자들의 신원을 포함하여 인증파일을 변경시킬것을 지령한다. 그런데 그 통보문을 사용자 F가 도중에서 가로 채여 통보문의 내용을 변경, 첨가, 삭제하고 다시 E에 보낸다. E는 관리자 D로부터 온것으로 생각하고 받으며 그대로 인증파일을 변경시킨다.
3. 통보문을 중간에서 꺼내보기보다 사용자 F는 필요한 기입으로 자기의 통보문을 만들고 그것을 관리자 D로부터 보내온것처럼 E에 전송한다. E는 그 통보문을 관리자 D로부터 온것으로 간주하고 그의 인증파일을 변경시킨다.

4. 어떤 직원이 경고없이 해고되였다. 관리자는 그의 등록자리를 무효로 하는 통보문을 봉사기체계에 보낸다. 무효가 설정되면 봉사기는 그 실행에 대한 확인으로서 그 직원의 파일에 통지를 보낸다. 그 직원이 통보문을 가로 챌수 있으며 중요한 정보들을 처리하는 봉사기에 제일 늦게 접근할수 있도록 상당한 기간 지연시킬수 있다. 그후에야 통보문이 전달되고 작용이 일어 나며 확인을 알린다. 직원의 행위는 상당한 기간은 알아 차릴수 없을수 있다.
5. 어떤 통보문이 고객으로부터 주식거간군에게 여러가지 트랜잭션처리에 대한 지령과 함께 보내진다. 그후 투자가 실패하면 고객은 그 통보문을 보낸것을 부인한다.

이러한 실례는 가능한 보안침입의 형태들을 다 보여 주지는 못하지만 망보안과 관련되는 내용들의 범위를 보여 준다.

인터네트보안은 매혹적이며 매우 복잡하다. 그 리유는 다음과 같다.

1. 망과 통신을 포괄하는 보안은 초학자들에게는 그리 단순하지 않다. 요구조건들이 복잡하지 않은것 같다. 즉 주요한 보안봉사에 대한 대부분의 요구들은 명백한 단어들인 기밀성, 인증, 비거부, 완정성으로 주어 질수 있다. 이 요구들에 부합되게 리용되는 꾸밈새들은 매우 복잡하며 그 리해에서 어느 정도 미묘한 추리가 있게 된다.
2. 특정의 보안꾸밈새 혹은 알고리듬을 개발하는데서 늘 가능한 대책들을 다 고려하여야 한다. 많은 경우 대책안들은 완전히 다른 각도에서 그 문제를 고찰하여 꾸밈새에서 예측 못했던 약점들을 찾아 내는 방법으로 구상한다.
3. 우와 같은 원인으로 하여 특정의 봉사들을 제공하기 위한 수속들은 흔히 반직관적이다. 이러한 정교한 대책을 필요로 하는가는 특정의 요구로부터는 명백치 않다. 각종의 대응수단들을 고려할 때만이 해당대책의 적용이 의미를 가질수 있다.
4. 각이한 보안꾸밈새를 설계한 다음에는 그것을 어디에서 리용하겠는가를 결심하는것이 필요하다. 이것은 물리적배치에서의 의미(즉 망의 어떤 점들에서 어떠한 보안꾸밈새들이 필요되는가)에서나 론리적의미(즉 TCP/IP와 같은 방식의 어떤 층이나 층들에 꾸밈새들이 놓이는가)에서 성립한다.
5. 보안꾸밈새는 흔히 몇개의 알고리듬 혹은 규약을 포함한다. 그것들은 또한 대방들이 어떤 비밀정보를 가질것을(실례로 비밀열쇠) 요구하는데 이때 그 비밀열쇠의 보호, 배송, 창조에 대한 문제가 제기된다. 여기에는 또한 그 보안꾸밈새를 개발하는 과제를 복잡하게 만들수 있는 통신규약에 대한 신뢰성문제도 있다. 실례로 보안꾸밈새의 기능개선이 송신자로부터 수신자에로의 통보문의 전송시간에 대한 시간제한을 요구한다면 가변적이고 예측할수 없는 지연을 가지는 임의의 규약 또는 망에서는 그러한 시간제한이 무의미해 질수 있다. 그러므로 고려해야 할 문제들이 많다.

이 장은 다음 장들에서 내용들을 구성하는 주제문제에 대한 일반적개괄을 준다. 여기서는 망보안봉사와 꾸밈새의 필요성을 야기시키는 공격의 형태에 대한 론의로부터 시작한다. 또한 보안봉사와 꾸밈새를 고찰할수 있는 일반적인 모형을 전개한다.

1.1 공격, 봉사 및 꾸밈새

어떤 기관의 보안요구에 성과적으로 접근하여 여러가지 보안제품과 방약을 선택하기 위하여 보안을 담당한 관리자는 보안요구들을 정의하고 그 요구를 만족시킬수 있는 방법들을 특징지울수 있는 어떤 체계적인 방법이 필요하게 된다. 하나의 방법은 다음과 같은 정보보안의 세 측면을 고찰하는것이다.

- **보안공격:** 어떤 기관이 가지고 있는 정보의 보안을 위태롭게 하는 임의의 공격
- **보안꾸밈새:** 보안공격으로부터 검출, 방어, 회복을 위해 설계된 꾸밈새
- **보안봉사:** 어떤 기관의 자료처리체계와 정보전송의 보안을 개선하는 봉사. 그 봉사는 보안공격의 방지를 목적으로 하며 또 그것들은 봉사를 제공하기 위한 하나이상의 보안꾸밈새들을 리용한다.

봉사

간단한 화제로부터 먼저 고찰하자. 정보보안봉사를 물리적문서와 표준적으로 결합된 기능형들을 재현하는것처럼 생각할수 있다. 상업, 외교, 군사작전, 개인들의 호상련계 등 각이한 령역에서 인간의 활동은 문서의 리용과 이 문서의 완정성에 대해 확신을 가지는 거래쌍방들에 의존된다. 문서들은 전형적으로 서명과 날자를 가진다. 그것들을 로출, 엿보기, 파괴로부터 보호해야 한다. 또 인증시키거나 증거로 될수 있으며 등록되거나 허락될수 있다.

정보체계들이 더욱 더 우리 사업에 보급되고 기본으로 되면서 전통적으로 종이문서로 수행되던 사업들을 전자정보가 대신하게 되였다. 따라서 종이문서와 결합된 기능의 형태들이 전자형태로 존재하는 문서우에서 수행될것이다. **전자문서**들은 다음과 같은 봉사나 기능을 갖추어야 한다.

1. 원래의 종이문서와 그 복사물사이의 차이가 있을수 있다. 그러나 전자문서는 다만 비트들의 렬이며 원본과 복사품사이에 아무런 차이도 없다.
2. 종이문서의 변경은 그 변경의 물리적증거를 얼마간 남길수 있다. 실례로 지우게 되면 연한 흔적이 남거나 종이에 보풀이 일게 된다. 콤퓨터에서 비트들의 변경은(혹은 신호에서 변경) 아무런 물리적흔적도 남기지 않는다.
3. 물리적문서와 관련한 증명과정은 대체로 그 문서의 물리적특성(례하면 손으로 쓴 수표나 도장)에 의존한다. 전자문서인증의 그러한 증명은 정보 그자체에 내재하는 내적증거에 기초하여야 한다.

표 1－1은 전통적인 문서와 전자문서 및 통보문사이의 류사한 기능의 일부를 보여준다. 이 기능들을 보안대책이 만족해야 할 요구로 볼수 있다. 표 1－1의 목록은 길며 그자체는 보안대책을 세우는데 지침으로 되지는 않는다. 콤퓨터나 망보안의 연구와 개발은 정보보안편의성에 요구되는 각이한 기능을 포괄하는 3～4개의 보안봉사에 집중된다.

표 1-1. 공통적인 **정보완정성기능**의 부분적목록[SIMM92b]

• 식별 • 인증 • 허가 및/혹은 확인 • 수표 • 증거(확증) • 일치 • 책임 • 령수 • 원문의 확인 및/혹은 령수	• 보증 • 접근(출입) • 정당성검증 • 출현시간 • 인증-프로그람 및/혹은 파일 • 투표 • 소유권 • 등록 • 승인/거부 • 개인성(비밀성)

보안봉사의 하나의 쓸모 있는 봉사는 다음과 같다.

- **기밀성:** 콤퓨터체계안의 정보와 전송된 정보는 인증된 대방이 읽기만을 위해서 접근할수 있다는것을 보증한다. 이러한 형태의 접근에는 인쇄, 현시 그리고 어떤 객체의 존재를 단순히 나타내는것을 비롯한 기타 로출형태들이 들어 있다.
- **인증성:** 전자문서 혹은 통보문의 원본은 그 신원이 거짓이 아니라는 보증밑에 정확히 일치한다는것을 보증한다.
- **완정성:** 인증된 대방들만이 콤퓨터체계의 자원과 전송된 정보를 변경할수 있다는것을 보증한다. 변경에는 전송된 통보문의 쓰기, 변경, 상태변경, 지연 및 재연이 속한다.
- **비거절성:** 통보문의 송신자와 수신자는 누구나 전송을 거절할수 없다.
- **접근조종:** 정보자원에 대한 접근은 목적체계에 의하여 조절될수 있어야 한다.
- **리용성:** 콤퓨터체계자원들은 필요할 때마다 인증된 대방들이 언제나 리용할수 있어야 한다.

꾸밈새

표 1-1에 제시된 모든 봉사를 제공하거나 모든 기능을 수행하는 단순한 꾸밈새는 없다.

이 책에서 취급한것과 같이 운영되는 많은 꾸밈새들의 변종을 볼수 있다. 현재 쓰이고 있는 보안꾸밈새의 기초에 놓여 있는 특정요소가 바로 암호기술이다. 암호화 즉 정보의 변환법과 같은 암호법은 보안을 제공하는 가장 공통적인 수단이다. 그러므로 이 책에서는 그러한 기술의 개발, 리용, 관리에 초점을 두었다(보안과 관련한 문헌들에서 소개되고 있는 대부분의 용어들에 대해서는 완전한 합의가 이루어 지지 않고 있다. 실례로 완정성은 때때로 정보보안의 모든 측면에 대해서 사용된다. 용어 인증은 때때로 완정성의 검증과 또는 다음에 제시되는 목록에서 완정성으로 제시된 각이한 기능에 관계되여 쓰이고 있다).

표 1-2. 속임의 형태([SIMM92b]참고)

1. 승인되지 않은 정보에 대한 접근(즉 비밀 혹은 개인성의 침범)
2. 책임을 회피하거나 다음의 목적을 위하여 다른 사람의 허가증을 리용하는 방법으로 다른 사용자로 위장
   ㄱ) 사기적인 정보의 조작
   ㄴ) 합법적인 정보의 변경
   ㄷ) 부정접근을 위하여 위조신원의 리용
   ㄹ) 트랜잭션에 대한 위조인증 혹은 위조보증
3. 사기군이 자기가 조작한 정보에 대한 의무와 책임의 부인
4. 사기군이 만들어 낸 정보를 다른 사용자로부터 받았다고 소송(즉 책임 혹은 의무의 기만적부여)
5. 보내지 않은 혹은 다른 시각에 보낸 정보를 수신자에게 보냈다고 소송
6. 실제로 수신하였지만 수신된것을 부인하거나 수신시각을 위조
7. 사기군의 합법적인 허가범위(접근, 창조, 배송 등)의 확장
8. 다른 사람의 허가(승인없이)를 변경한다(허위적인 다른 사람을 등록, 허가설정기간의 확장, 한정 등).
9. 어떤 정보를(공개통신에서) 다른 정보에(은밀한 통신) 숨기는것
10. 능동(검출 안된)중계점으로서 다른 사용자들사이의 통신련결선들에 끼여 든다.
11. 정보자체가 은폐되였더라도(통신통로로부터 자료기지까지의 자료분석의 일반화, 쏘프트웨어 등) 누가 어느 정보(원천,파일들 등)에 언제 접근하는가를 알려고 한다.
12. 사기군이(규약의 내용상으로) 비밀을 지키고 있는것처럼 가정되는 정보로출에 의해 정보완정성규약이 파괴된다.
13. 어떤 은밀한 기능을 첨가하여 쏘프트웨어의 기능이 제대로 발휘 못되게 한다.
14. 어떤 규약에 부정확한 정보를 침투시키는 방법으로 다른 사용자들이 그것을 위반하게 한다.
15. 체계에서 큰 실패가 일어 나게 함으로써 그 규약에 대한 신뢰성에 손상을 준다.
16. 다른 사용자들사이의 통신을 막으며 특히 비밀리의 간섭으로 인증통신이 그렇지 않은것으로서 거부하게 한다.

공격

지.제이.시몬(G.J.Simmons)이 지적한것처럼 정보보안은 사기행위를 어떻게 막는가 또는 그것이 실패하면 정보 그자체가 의미 있는 물리적실체를 가지지 않는 정보적체계에서 사기를 검출하는 방법이다.

우의 표 1-2는 더 명백한 일부 사기들에 대한 실례들을 보여 주는데 그러한것들은 현실에서 많이 나타나고 있다. 여기에는 어떤 기관이나 개인(직원들을 대표하는 어떤 기관)들이 대처해야 할 여러가지 형태의 공격들이 있을수 있다. 기관이 관여할수 있는 공격들의 성격은 그때그때의 형편에 따라 많이 달라 진다.

부닥칠수 있는 공격의 일반적인 형태들을 고찰하면서 각이한 각도에서 문제를 연구할수 있다. 이것이 다음 절의 주제이다.

1.2 보안공격

콤퓨터체계 혹은 망의 보안에 대한 공격들은 콤퓨터체계의 기능을 정보를 제공하는 것으로 고찰하여 잘 특징지을수 있다. 일반적으로 원천지(주기억구역 혹은 파일 등)로부터 목적지(다른 파일 혹은 다른 사용자)에로의 정보흐름이 있게 된다.

이 표준흐름을 그림 1-1의 ㄱ에서 보여 주었다.

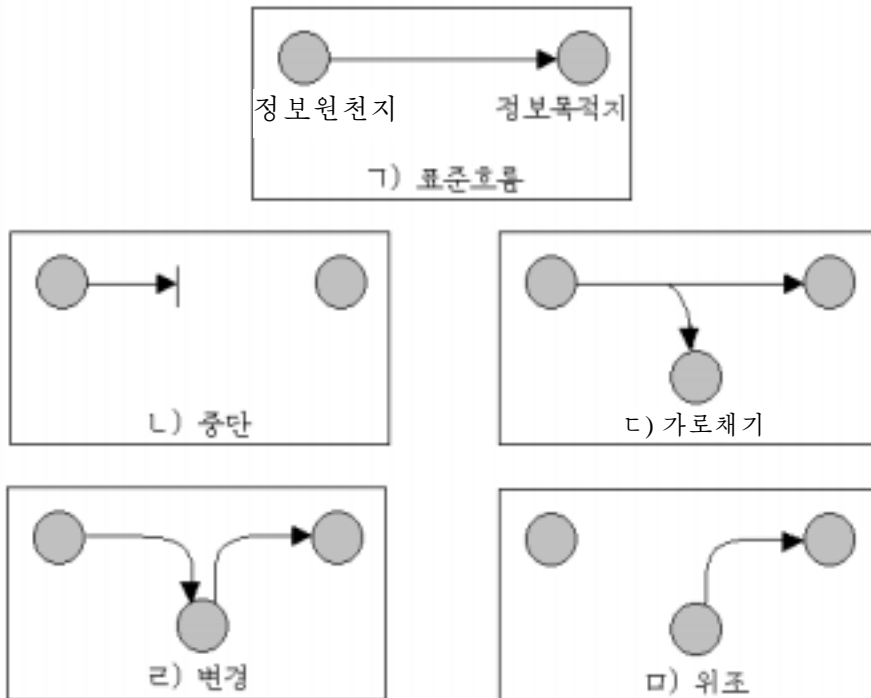



그림 1-1. 보안위협

이 그림의 나머지부분들에서 다음 4개의 공격의 일반적범주들을 보여 주고 있다.

- **중단:** 체계의 자원이 파괴되며 혹은 리용할수 없게 되거나 무효하게 된다. 이것은 유용성에 대한 공격이다. 이러한 실례들에는 어떤 하드웨어부분의 파괴(하드디스크와 같은), 통신선의 절단 또는 파일관리체계의 무력화 등이 포함된다.
- **가로채기:** 인증되지 않은 자들이 자원에 대한 접근을 얻는다. 이것은 기밀성에 대한 공격이다. 인증을 얻지 못한 대상들로는 사람, 프로그람 혹은 콤퓨터가 될수 있다. 실례로 어떤 망에서 자료를 획득하기 위한 도청 혹은 프로그람이나 파일들의 부정복사이다.
- **변경:** 어떤 부정침입자(인증되지 않은자)들이 자원에 접근할뿐아니라 그 자원

을 조절하는 공격이다. 이것은 완정성에 대한 공격이다. 실례로서 어떤 자료 파일에서 값들을 변경하거나 어떤 프로그람이 잘못 실행되도록 변경 혹은 어떤 망에서 전송되는 통보문을 변경시키는것 등이다.

- **위조:** 어떤 부정침입자가 체계에 모조객체를 삽입하는 공격이다. 이것은 **인증에 대한 공격**이다. 실례로 파일에 레부호들을 첨가하거나 어떤 망에 위조정보들을 삽입하는것 등이다.

이 공격들을 일반적으로 소극적(피동)공격과 적극적(능동)공격으로 분류할수 있다 (그림 1-2).

우와 같은 공격에 대한 분류는 스테레 켄트(Stere kent)[KENT77]가 처음으로 제기하였다. 이 분류는 지금도 가치가 있으며 대부분의 보안공격에 대한 서술에서 기초로 되고 있다.

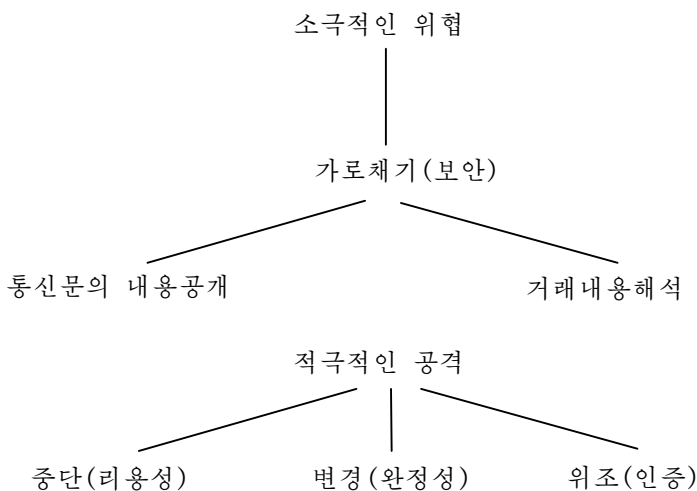



그림 1-2. 소극적 및 적극적망보안위협

소극적공격

소극적공격은 본질상 전송정보를 감시하고 도청하는것이다. 적의 목적은 전송되는 정보를 얻는것이다. 소극적공격의 두 형태는 통보문의 내용을 공개시키거나 거래되는 내용을 해석하는것이다.

통보문의 내용공개는 쉽게 리해할수 있다. 전화대화, 전자우편통보문, 전송파일에 민감한 정보 혹은 비밀정보가 포함될수 있다. 적이 이 전송내용들을 알지 못하게 해야 한다.

두번째 소극적공격인 **거래해석**은 더 은밀하다. 통보문의 내용이나 기타 정보거래내용을 감추는 방법이 있다면 비록 적들이 통보문을 포착하였다 하더라도 그 통보문으로부터 정보를 꺼낼수 없을것이다. 정보의 내용을 감추는(음페하는) 공통적기술이 암호이다. 암호보호를 적당히 한다면 적은 이 통보문의 패턴을 주시하면서 통신하는 가입자들의 이름과 위치를 알아 내고 교환되는 통보문의 길이와 주파수를 얻을수 있다. 이 정보는 진행되고 있는 통신의 성질을 추측하는데 쓸모가 있을수 있다.

소극적공격들은 그 자료에 대해 아무런 변경도 주지 않기때문에 검출하기가 매우 어렵다. 그러나 이 공격들을 막을수 있다. 소극적공격을 다루는데서는 검출보다 오히려 예방이 중요하다.

적극적공격

공격의 두번째 범주는 적극적인 공격이다. 이 공격들은 자료흐름의 일정한 변경 또는 허위흐름의 창조를 포함하는데 4개의 범주 즉 가장, 재연, 통신문의 변경, 봉사의 제거 등으로 구분할수 있다.

가장은 한 실체를 다른 실체로 꾸밀 때 발생한다. 가장공격은 적극적인 공격의 다른 하나의 형태를 포함한다. 실례로 인증렬은 어떤 유효인증렬이 일어 난후에 포착되고 재현될수 있으며 그래서 적은 특권을 가진 어떤 인증된 실체가 허위적으로 추가특권들을 가지도록 할수 있다.

재연은 어떤 자료단위의 소극적인 포착과 그것을 재전송함으로써 인증되지 않은 효과를 초래하는것이다.

통보문의 변경은 본래의 통보문의 어떤 부분이 변경되였거나 그 통보문이 지연 또는 재지령되여 비인증효과를 나타내는것이다. 실례로 통보문 《존 스미스는 극비파일문서를 읽을것을 허락한다.》를 《프레드 브라운은 극비파일문서를 읽는것을 허락한다.》로 수정하는것 등이다.

봉사거절은 통신시설의 일반적리용 또는 관리를 방해하거나 금지시킨다. 이 공격은 특수한 목적을 가질수 있다. 실례로 어떤 실체는 특정의 목적지에로 가는 모든 통보문들을 막아 버릴수 있다(즉 비밀검열봉사). 봉사거절의 다른 형태는 망의 성능을 떨구기 위하여 그것을 무능력하게 하거나 통보문과부하를 걸어 망전체의 혼란을 일으키는것이다.

적극적공격은 소극적공격과 반대되는 특성을 가진다. 소극적공격은 그 검출이 어려운 반면에 그것을 검출하면 대책을 취하여 막을수 있다. 한편 적극적공격은 절대적으로 막는것이 어려운데 그것은 항상 통신시설과 경로들을 물리적으로 보호하여야 하기때문이다. 대신 그것을 검출하고 그것에 의해 생겨 난 와해와 지연으로부터 회복시키는것이 목적으로 된다.

1.3 보안봉사

기밀성

기밀성(confidentiality)은 소극적공격으로부터 전송되는 자료의 보호이다. 통보문 내용의 공개에 대해서는 여러개의 보호수준들로 갈라 볼수 있다. 가장 폭 넓은 봉사는 일정한 기간에 두 사용자사이에 전송되는 모든 자료를 보호하는것이다. 실례로 어떤 가상회선이 두 체계사이에 설치되면 이 넓은 보호는 가상회선상에서 전송되는 임의의 사용자자료의 공개를 방지한다. 이 봉사의 보다 좁은 형식들은 어떤 하나의 통보문 지어는 어떤 통보문안에서 지적된 마당들의 보호 등으로 정의되고 이러한 형식은 넓은 보호보다 쓸모가 적으며 또한 실현이 더 복잡하고 비용이 많이 들수 있다.

기밀성의 다른 측면은 해석으로부터 자료흐름을 보호하는것이다. 이것은 통신설비에서 거래의 원천지, 목적지, 주파수, 길이 등의 특성들을 공격자가 관측할수 없게 할것을 요구한다.

인증

인증(authentication)봉사는 통신이 인증된다는 보증과 관련된다. 단순한 통보문의 경우(즉 경보나 경보신호와 같은)에 인증봉사의 기능은 그 통보문이 신용할만한 원천지로부터 왔다는 확인을 보증하는것이다. 어떤 말단으로부터 주콤퓨터에로의 접속과 같은 실행중의 대화인 경우에는 다음의 두 측면들이 포함된다. 첫째로, 련결초기에 봉사는 두 실체들이 인증되였다는것을 보증한다(즉 매 실체는 서로 보증한다). 둘째로, 봉사는 련결이 3자가 비인증송신과 수신을 목적으로 두 합법적대방의 하나처럼 속이는 어떠한 방법으로도 간섭할수 없다는것을 보증하여야 한다.

완정성

기밀성과 마찬가지로 완정성(integrity)은 통보문들의 흐름, 단일통보문 또는 어떤 통보문의 선택된 마당들에 적용할수 있다. 가장 쓸모 있고 간단한 수법은 전체 통보문흐름의 보호이다.

련결지향완정성봉사는 그 통보문의 취급이 재연, 재배렬, 변경, 삽입, 복제가 없이 보낸 그대로 받는다는것을 보증한다. 자료의 파괴도 역시 이 봉사의 대상이다. 따라서 련결지향완정성봉사는 봉사의 거절과 통보문흐름의 변경을 다 대상한다. 한편 내용의 크기에 관계없이 다만 개별적인 통보문만 취급하는 련결 없는 완정성봉사는 일반적으로 통보문의 변경만을 못하게 하는 보호를 제공한다. 켄트(Kent)는 재연, 재배렬을 일부 방지하면서도 엄격한 순차를 요구하지 않는 응용에 혼성봉사를 써먹을수 있다는것을 지적하였다[KENT93a].

여기서 봉사를 회복할수 있는것과 없는것으로 구별할수 있다. 완정성봉사는 적극적 공격에 관계되기때문에 예방보다 검출에 주의를 돌린다. 완정성에서 위반이 검출되면 봉사는 그 위반에 대하여 통보만 하므로 쏘프트웨어의 일부 다른 부분이나 사람의 간섭에 의하여 그 위반을 회복하여야 한다. 자료의 완정성손실을 회복하는데 리용되는 꾸밈새가 있다. 자료회복꾸밈새들의 병합은 일반적으로 더 매혹적인 대안이다.

비거절

비거절은 송신자나 수신자가 전송된 통보문을 거절(repudiation)하는것을 예방한다. 즉 어떤 통보문을 보내면 수신자는 그 통보문이 자기의 대방이 보낸것이라는것을 확신할수 있다.

접근조종

망보안과 관련하여 접근조종(Access Control)은 통신련결을 거쳐 주콤퓨터체계와 응용프로그람들에 대한 접근을 조종, 제한하는 능력이다. 이러한 조종을 달성하기 위해 접근을 얻으려는 매 실체는 우선 신원확인되거나 인증되여야 하며 따라서 접근권리는 개별적으로 부여된다.

리용성

일부 공격들에 의해 리용성(availability)이 상실 혹은 적어 질수 있다. 이 공격들의 일부는 인증보안이나 암호와 같은 자동적인 대응수단에 의해 방지되지만 다른것들은 분산체계요소들의 리용성의 상실로부터 회복 혹은 보호하기 위하여 몇가지 물리적작용을 요구하는 경우도 있다.

1.4 호상련결망보안모형

여기서 대부분 론의하게 되는 모형이 그림 1-3에 일반적형태로 주어 졌다. 통보문은 호상련결망을 통하여 한 대방으로부터 다른 신용하는 대방에게 전송된다. 이 트란잭션에서 당사자들인 쌍방은 정보교환을 위해 협력하여야 한다. 론리적정보통로는 호상련결망에서 원천지로부터 목적지까지의 경로정의와 두 당사자들에 의한 통신규약(실례로 TCP/IP)의 공동리용에 의하여 설정된다.

보안문제는 기밀성, 인증 등에 위협을 줄수 있는 적으로부터의 정보전송보호를 필요로 할 때 제기되게 된다. 모든 보안기술에는 두개의 성분이 포함된다.

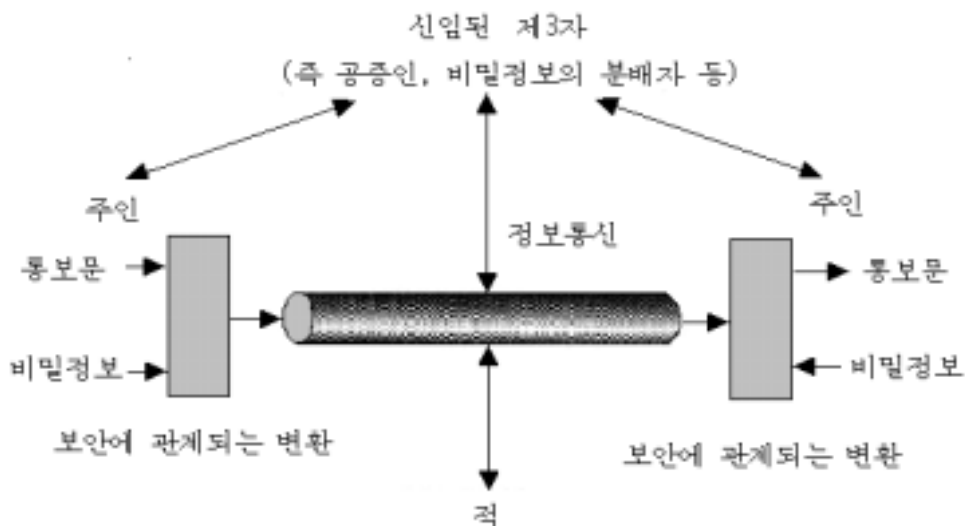



그림 1-3. 망보안모형

- 보내려는 정보에 대한 보안련관변환. 실례로 적이 통보문을 읽을수 없도록 혼란시키기 위한 통보문의 암호화와 송신자의 신원을 확인하는데 리용할수 있는 통보문의 내용에 기초한 어떤 부호의 첨부를 들수 있다.
- 당사자(송신자, 수신자)들이 공유하는 적에게 로출되지 말아야 할 어떤 비밀정보, 실례로 송신전에는 통보문을 쪼각화하고 수신하였을 때에는 다시 통합하는데 리용되는 암호화열쇠를 들수 있다.

신용되는 제3자는 비밀전송을 요구한다. 실례로 그는 적으로부터 비밀정보를 지키면서 두 당사자에게 비밀정보를 배송하는 임무를 수행할수 있다. 혹은 어떤 통보문전송의 인증과 관련한 두 가입자사이에 론쟁을 중재해야 할수도 있다.

이 일반적모형은 특정한 보안봉사의 설계에서 다음의 4개의 기본과제가 제기된다는 것을 보여 준다.

1. 보안에 관계되는 변환을 수행하는 알고리듬을 설계한다. 알고리듬은 적이 그 목적을 방해할수 없도록 되여야 한다.
2. 알고리듬과 함께 리용되는 비밀정보를 생성한다.
3. 비밀정보를 배송 및 공유할수 있는 방법을 개발한다.
4. 특정의 보안봉사를 위해 보안알고리듬과 비밀정보를 리용하는 두 당사자(송신자, 수신자)들이 리용할 규약을 지정한다.

이 책의 많은 내용이 그림 1-3에서 보여 준 방식에 알맞는 보안봉사와 보안꾸밈새들에 바쳐 지고 있다. 그러나 꼭 그런 방식은 아니지만 흥미 있는 다른 보안련관문제들도 고찰하였다. 그러한 보안모형을 그림 1-4에 보여 주었는데 그것들은 바람직하지 못한 접근으로부터 어떤 정보체계를 보호하기 위한것이다. 대부분의 독자들은 해커와 관련한 문제들과 구면일수 있는데 해커는 어떤 망우에서 접근할수 있는 체계에 침투하려고 한다. 해커들중에는 나쁜 의도가 없이 그저 어떤 콤퓨터체계에 침투하거나 중단시키는것으로 만족하는 자도 있고 또 파괴를 목적으로 하는 나쁜 침입자도 있을수 있으며 또한 돈을 바라고 콤퓨터자원에 침투하는 범죄자(즉 신용카드번호를 얻거나 비법적인 현금전송을 수행하는)도 있을수 있다.

다른 위험한 접근형태는 체계의 부족점을 리용하고 응용프로그람이나 편집기, 콤파일러와 같은 편의프로그람에 영향을 줄수 있는 론리가 콤퓨터체계에 들어 가는것이다.

- **정보접근위협** 그 자료에 대한 접근을 하지 말아야 할 사용자로서 자료를 변경 혹은 가로채는것
- **봉사위협** 합법적인 사용자의 리용을 금지시키기 위해 콤퓨터에서 봉사결함들을 리용한다.

비루스와 웜은 쏘프트웨어공격의 대표적인 두가지 실례이다. 이러한 공격들은 다른 목적의 쏘프트웨어에 숨은 론리체계가 들어 있는 디스크를 통해 체계에 들어 가게 된다. 그것들은 망을 통하여 어떤 체계에 들어 갈수 있다. 이 후자가 망보안에 더 관계된다.

불리한 접근에 대처하는데 필요되는 보안기구는 크게 두 범주로 갈라 진다(그림 1-4를 보시오). 첫 범주는 문지기기능이라고 부른다. 거기에는 인증된 모든 사용자들에

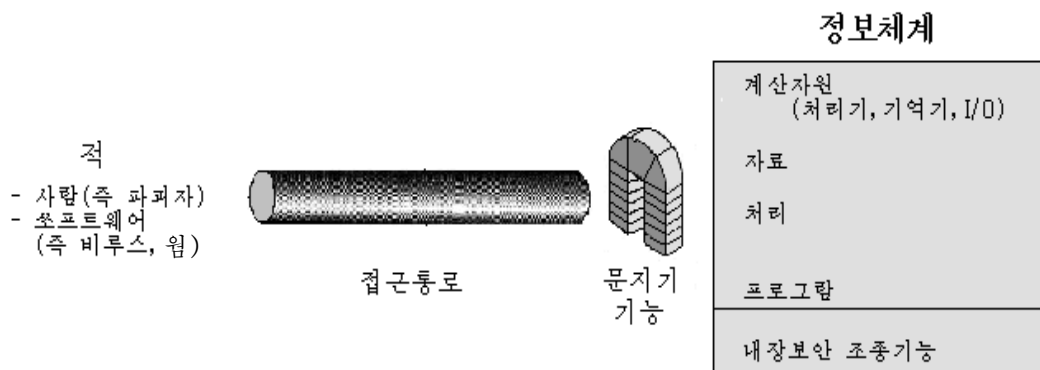



그림 1-4. 망접근보안모형

대한 접근을 부정하도록 설계된 통과암호식가입수속과 **웜**과 비루스 기타 류사한 공격들을 탐색하도록 설계된 보호론리체계가 들어 있다. 일단 원하지 않는 사용자나 원하지 않는 쏘프트웨어의 접근이 성사되면 여러 **내부조종**으로 이루어 진 두번째 방어선은 원하지 않는 침입자들의 존재를 검출하기 위해 기억된 정보를 해석하는 활동을 감시한다.

1.5 이 책에 대한 개괄

이 장에서는 책전반내용에 대한 소개를 주었다. 나머지 장들의 개요는 다음과 같다.

전통암호화: 고전기술

망과 통신보안에서 가장 중요한 자동화도구가 암호화이다. 암호화에는 두가지 형이 있다. 즉 전통 혹은 대칭암호화와 공개열쇠 혹은 비대칭암호화인데 2장에서는 고전의 전통암호기술을 취급하였다. 이 장에서는 암호학과 암호분석학에 대한 친절하고 흥미 있는 소개를 주면서도 중요한 개념들에 대해서는 특별히 강조하였다.

전통암호화: 현대기술과 알고리듬

3장에서는 가장 널리 쓰이는 암호기술인 자료암호표준(DES)을 중심으로 하여 현대대칭암호학의 원리를 소개하였다. 이 장에서는 대부분의 현대전통암호화방식의 기본구조로 되는 페이스텔(Feistel)암호를 소개하고 암호분석과 설계문제에 대하여 취급하였다.

4장은 일부 중요한 현대블로크암호알고리듬들을 비롯하여 론의범위를 확장하였다.

전통암호화리용의 기밀성

전통암호화알고리듬의 실제적구조를 취급하는 문제외에 많은 설계문제들이 기밀성을 보장하기 위한 전통암호화의 리용과 관련되여 있다. 5장에서는 그중에서 중요한 내용들에 대하여 개괄한다. 이 장에서는 말단대말단련결암호화의 론의, 거래기밀성을 달성하기 위한 기술, 열쇠배포기술을 취급하였다. 관계되는 중요한 화제로 란수발생도 서술되고 있다.

공개열쇠암호화

전통암호화와 다른 중요한 암호화가 공개열쇠암호화이다.

그것은 통신보안에서 혁신이다. 6장에서는 공개열쇠암호화에 대하여 소개하고 그 리용에서 기밀성을 보장하는데 기본을 두었다. Rivest-Shamir-Adleman(RSA)알고리듬이 구체적으로 심의되였으며 열쇠관리의 문제도 재고찰되였다. 이 장에서는 또한 널리 쓰이고 있는 디피-헬만(Diffie-Hellmann)의 열쇠교환기술과 타원곡선에 기초한 최초의 공개열쇠방법도 고찰하였다.

수론에 대한 입문

대부분의 공개열쇠방식은 수론에 기초하고 있다. 독자들이 수론의 결과들을 확신하게 되면 그 개념들을 기본적으로 파악할수 있다. 7장에서는 그 개념들을 명백히 하기 위한 많은 실례들과 개괄을 주었다.

인증과 하쉬함수

보안대책으로서 기밀성과 마찬가지로 중요한것이 인증이다. 최소한 통보문의 인증은 어떤 통보문이 지정된 송신자로부터 왔다는것을 보증한다. 게다가 인증은 변경, 지연, 재연, 재배렬에 대한 보호를 포함한다. 3장에서는 인증의 필요성에 대한 해석으로부터 시작하여 인증방법에 대한 구체적인 서술을 주고 있다. 인증방식의 관건적요소는 인증자의 리용인데 흔히 통보문인증부호(MAC) 혹은 하쉬함수를 인증자로 쓴다. 이런 형들의 알고리듬들에 대한 설계를 고려하면 몇개의 특수한 실례들이 해석된다. 9장에서는 현재 가장 중요하게 쓰이는 하쉬함수들과 HMAL로 알려 진 인터네트표준 MAC 등에로 론의범위를 확장하였다.

수자서명과 인증규약

인증의 중요한 형태는 수자서명이다. 10장에서는 수자서명을 생성하는데 쓰이는 기술을 고찰하고 중요한 표준 즉 수자서명표준(DSS)에 대하여 서술하였다.

수자서명에 기초한 각이한 인증기술들은 인증알고리듬과 함께 블로크로 작성되고 있다. 그러한 알고리듬의 설계에는 많은 명백한 안전규약들을 파괴할수 있는 교묘한 공격들에 대한 해석을 포함한다.

인증응용

11장에서는 현재 리용되고 있는 가장 중요한 인증명세서중에서 2개를 개괄하였다. Kerberos는 광범히 응용되는 전통암호화에 기초한 인증규약의 하나이다.

X.509는 어떤 인증알고리듬을 지정하며 어떤 확인편의도구들을 정의한다. 확인편의도구는 사용자공동체가 그 공개열쇠의 타당성에 확신을 가지게끔 공개열쇠에 대한 확인을 얻을수 있게 한다. 이 편의도구들은 많은 응용들에서 블로크를 조정하는것으로서 리용된다.

전자우편보안

가장 많이 리용되는 응용이 전자우편이다. 전자우편편의도구부분으로 인증 및 비밀봉사를 보장하려는 요구가 높아 지고 있다. 12장에서는 가까운 장래에 전자우편보안을 지배하리라고 보아지는 두가지 방법을 고찰하였다. PGP는 기관이나 정보에 관계없이 광범히 리용되는 방식이다. 따라서 기관들이 운영하는 망구조에 병합되려는 개별적사람들에게도 적합하다. S/MIME(Secure/Multipurpose/Internet Mail Extension)는 전형적인 인터네트표준으로 개발되였다.

IP보안

인터네트규약(IP)은 인터네트와 전용인트라네트에서 중심적요소이다. 따라서 IP준위에서 보안은 임의의 인터네트기반의 보안방식의 설계에서 중요하다. 13장에서는 다음세대 IP와 현재의 IP를 다 운영하게끔 개발된 IPv6이라고 하는 IP보안방식을 고찰한다.

Web보안

전자상거래용으로 WWW리용의 급격한 장성과 정보의 보급은 Web에 기초한 강한

보안의 필요성을 제기하였다. 14장에서는 이 중요하고도 새로운 보안령역에 대한 개괄을 주었다. 그리고 2개의 열쇠표준들 즉 안전스케트층(SSL)과 보안전자트랜잭션(SET)을 고찰한다.

침입자, 비루스, 웜

15장에서는 망에 기초한 처리체계에서 약점을 리용하는 프로그람과 해커에 의해 존재하는 봉사위협들과 정보접근의 각이한 형태들을 서술하였다. 이 장에서는 인증되지 않은 사용자 혹은 침입자에 의한 공격의 형태에 대한 론의로부터 시작하여 예방과 검출의 각이한 수법들을 분석하였다. 그리고 비루스를 비롯하여 프로그람적인 위협들에 대해서도 론의하였다.

방화벽

외부의 위협으로부터 국부적인 콤퓨터자원을 보호하기 위한 표준적인 수법은 방화벽의 리용이다. 16장에서는 방화벽설계의 원리에 대하여 론의하고 전용기술을 고찰하였다.

참고문헌

[PFLE97]에 콤퓨터와 망보안에 대하여 구체적으로 소개되여 있다. 또 다른 중요한 문제는 [ARA95]에서 취급되였다.

ABR95. Abrams, M: Jajodia, S; and Podell, H.,eds. *Information Security:An Integrated Collection of Essays*. Los Alamitos, CA : IEEE Computer Society Press, 1955.

PFLE 97 Pfleegr,C Security in Computing. Upper Saddle River, NJ: Prentice Hall. 1997

부록 1: 인터네트와 Web자원

인터네트와 Web에는 이 책의 내용을 리해하고 해당 분야에서의 새로운 개발들을 계속 받아 들이는데서 도움이 될수 있는 자원들이 많다.

이 책의 Web싸이트

이 책의 전용 Web페지는 http//www.shore.net/~ws/Security2e. html에 설치되였다. 이 싸이트에는 다음과 같은것들이 포함된다.

- 부록에 주어 진 목록을 비롯한 다른 Web싸이트에 대한 련결들은 Web상의 관계되는 자원들에 대한 관문을 제공한다.
- 직결식투명한 이 책의 주인은 PDF(adobe Acrobat)형식으로 이 책의 대부분

의 도형들에 대해서 직접 명확히 리해할수 있다.
- 게재된 정보는 책의 인터네트우편목록용으로 제공된다.
- 책에 기초한 과정안을 위한 홈페지에로의 련결도 포함시켰다. 이 페지들은 다른 교원들이 자기 과정안을 구성하기 위한 안을 세우는데서 도움이 될수 있다.

이 책에서 인쇄나 기타 오유를 발견하면 곧 이 책에 대한 고침표를 http://ww.shore.net/~ws에서 리용할수 있다. 파일은 필요하면 갱신할수 있다.
오유를 발견하면 ws@shore. net로 전자우편을 보내시오. 저자의 다른 책들에서의 고침표들도 같은 Web싸이트에 있으므로 그 책들에 대한 주문정보의 량을 줄인다.

다른 Web싸이트

이 책의 주제와 관련되는 몇가지 정보를 제공하는 많은 싸이트들이 있다. 몇가지 실례를 보면 다음과 같다.
- COAST: 암호학과 망보안에 관계되는 련결들의 종합적인 모임
- IETF Security Area: 인터네트보안규격화에서 이룩된 최신 성과들
- Computer and Network Security Reference Index: (콤퓨터와 망보안참조색인)상품들과 판매자, FAQ들, 새소식그룹기록, 론문들, 기타 다른 Web싸이트에 대한 풍부한 색인
- The cryptography FAQ: 암호학의 모든 측면을 포괄하는 풍부하고도 쓸모있는 FAQ
- Tom Dunigan's Security page: 암호학과 망보안Web싸이트에 대한 지적자들의 목록
- IEEE Technical Committee on Security and privacy: IEEE의 활동과 관련한 정보, 편지들을 복사한다.

다음 장들에서는 보다 전용적인 Web싸이트들을 참고문헌에 대한 절들에서 소개한다.

USENET Newsgroups

많은 전자신문(USENET news)그룹들은 망보안과 암호학의 일부 측면들에 대하여 밝히고 있다.거의 모든 전자신문그룹들과 마찬가지로 여기에도 높은 잡음대 신호비 (N/S)가 있다. 그러나 요구에 부합되면 시험해 볼 필요가 있다. 그중 관계되는것들은 다음과 같다.

- Sci.crypt: 암호학과 관계되는 화제들에 대한 일반적론의
- Sci.crypt.research: 이것은 연구화제들을 취급하는 적절한 신문그룹이다. 게시된 내용들은 암호학의 기술적측면과 일정한 관계가 있다.
- Alt.security: 보안화제에 대한 일반적론의
- Comp.Security.firewalls: 방화벽들과 그 기술에 대한 일반적론의

제1편. 전통암호

제2장. 전통암호: 고전기술

대칭암호 또는 단일열쇠암호라고도 불리우는 전통암호는 공개열쇠암호가 나오기전의 유일한 암호였다. 이것은 가장 널리 쓰이고 있는 암호의 두 형태들중의 하나로 되고 있다. 이 장과 다음 두 장에서는 각이한 전통암호알고리듬들에 대하여 고찰한다. 그리고 5장에서는 전통암호에서 열쇠의 리용과 관련한 몇가지 문제들을 고찰한다.

먼저 전통암호에 대한 일반모형을 고찰하고 그다음 콤퓨터시대 이전에 리용하였던 각이한 알고리듬들을 고찰한다. 3장에서는 현재 광범히 리용되고 있는 암호알고리듬인 DES에 대하여 취급한다.

2.1 전통암호모형

그림 2-1에서 전통암호의 처리과정을 보여 주었다. **평문**이라고 부르는 의미 있는 본래의 통보문은 **암호문**이라는 겉보기에 무질서하고 무의미한것으로 변환된다. 암호화처리는 알고리듬과 열쇠로 이루어 진다. 열쇠는 평문과 독립인 값이다. 알고리듬은 리용되는 특정의 열쇠에 따라서 각이한 출력을 낸다. 열쇠를 변화시키면 그 알고리듬의 출력이 달라 진다.

일단 암호문이 생성되면 그것을 전송할수 있다. 수신되면 그 암호문은 암호화에 쓰였던것과 같은 열쇠와 복호알고리듬을 리용하여 원래의 평문으로 다시 변환된다.

전통암호에 의한 보안은 여러개의 인자(요인)들에 의존된다. 첫째로, 암호알고리듬은 그 암호문에만 기초하여 통보문을 분석할수 없을만큼 충분히 강하여야 한다. 또한 전통암호에 의한 보안은 알고리듬의 비밀이 아니라 열쇠의 비밀에만 의존한다. 즉 암호문과 암호알고리듬/복호알고리듬에 대한 지식에 기초하여 통보문을 분석하는것은 비현실적이라고 본다. 다시말하여 알고리듬에 대한 비밀은 필요 없고 열쇠의 비밀만 필요하다.

전통암호의 이러한 특성은 그것을 광범히 쓸수 있게 하였다. 알고리듬을 비밀보관할 필요가 없다는 사실은 설계자들이 자료암호알고리듬의 저가격의 소편실장을 개발할수 있다는것을 의미한다. 이 소편들은 많은 제품들에 결합되여 광범히 리용되고 있다. 전통암호의 리용에서 기본은 열쇠의 비밀을 지키는것이다.

그림 2-2를 통하여 전통암호방식의 기본요소들을 좀 더 구체적으로 보자. 원천지는 평문 $X=[x_1, x_2,\cdots,x_M]$으로 어떤 통보문을 만든다. X는 어떤 유한자모모임의 문자들의 렬이다. 전통적으로 문자는 흔히 26개의 대문자로 이루어 져 있다. 오늘날에는 2진자모 {0, 1}이 일반적으로 쓰이고 있다.

암호화를 위해 $K=[K_1, K_2,\cdots, K_j]$형식의 어떤 열쇠를 생성한다. 만일 열쇠가 통보문 원천지에서 생성되면 그것은 어떤 안전한 통로를 통하여 목적지에 제공되여야 한다. 다른 방법으로는 제3자가 열쇠를 생성하고 그것을 원천지와 목적지에 안전하게 배달할수 있다.

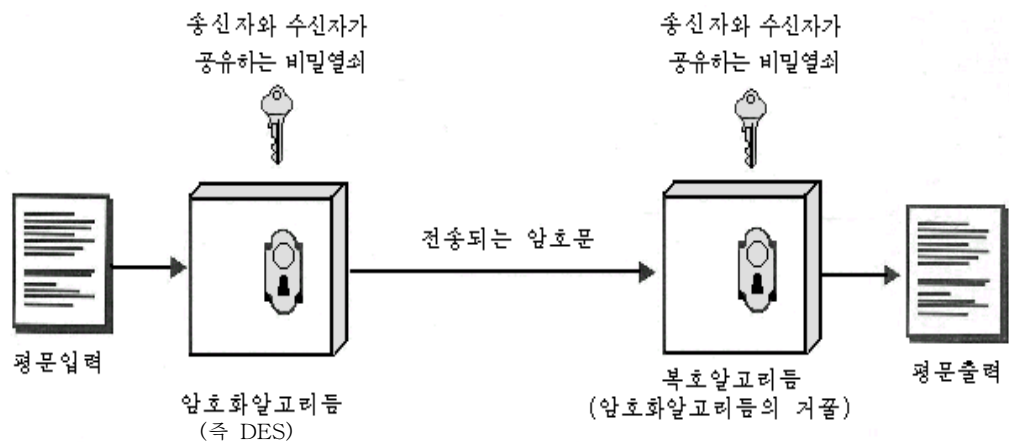



그림 2-1. 전통암호의 간단한 모형

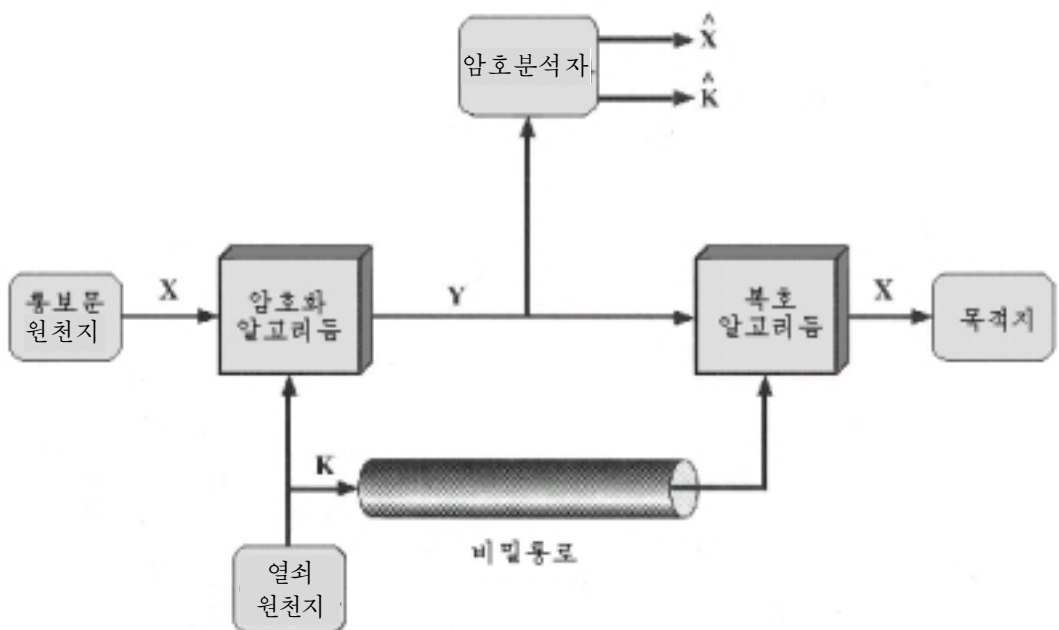



그림 2-2. 전통암호의 모형

암호화알고리듬은 입력으로서 통보문 X와 암호열쇠 K를 가지고 암호문 $Y=[Y_1, Y_2, \cdots, Y_N]$을 만들어 내는데 이것은 다음과 같다.

$$Y=E_K(x)$$

이 표기는 열쇠 K에 의하여 결정되는 지정된 함수와 평문 X의 함수로서 암호화알고리듬 E를 리용하여 Y가 생성된다는것을 반영한다.

같은 열쇠를 가지고 있는 약속된 수신자는 전송자료에 대하여 거꿀변환을 할수 있다.

$$X=D_K(Y)$$

K나 X에 대한 접근을 얻지 못한 적은 Y를 관측하여 K나 X 또는 X와 K를 다 회복하려고 시도한다. 적이 암호화알고리듬과 복호알고리듬을 알고 있다고 가정하자. 만일 적이 이 특정의 통보문에만 관심이 있으면 얻어 진 평문 $\hat{X}$를 평가하여 X를 회복하려고 할것이다. 물론 적은 앞으로의 통보문도 알려고 할수 있다. 이런 경우에 얻어 진 $\hat{K}$로부터 K를 회복하려고 한다.

암호학

암호체계를 일반적으로 세개의 독립적인 요소들로 분류한다.

1. **평문을 암호문으로 변환하는데 쓰이는 연산들의 형태.** 모든 암호알고리듬들은 다음과 같은 2개의 일반적원리에 기초하고 있다. **대입**-이것은 평문에 있는 매 요소(비트, 비트렬, 문자)에 다른 원소를 대응시킨다. **치환**-평문에 있는 원소들을 재배렬한다. 기본요구는 정보가 잃어 지지 않게 하는것이다(즉 모든 연산은 가역적이여야 한다). 합성체계라고 부르는 체계들은 대입과 치환의 여러 단계를 포함한다.
2. **리용되는 열쇠들의 수.** 송신자와 수신자가 같은 열쇠를 리용한다면 그 체계는 대칭 혹은 단일열쇠암호 또는 비밀열쇠암호나 전통암호라고 부른다. 만일 송신자와 수신자가 각각 서로 다른 열쇠를 리용한다면 그 체계는 비대칭 또는 두열쇠 또는 공개열쇠암호라고 부른다.
3. **평문을 처리하는 방식. 블로크암호**는 한번에 한개 입력블로크를 처리하여 매 입력블로크에 대하여 하나의 출력블로크를 생성한다. 흐름암호는 입력을 련속적으로 처리하여 한번에 하나의 원소씩 생성하여 간다.

암호분석

X 혹은 K 또는 둘 다 발견해 내려는 처리를 암호분석이라고 부른다.

암호분석에 쓰이는 전략은 암호방식의 성질과 암호분석자가 리용할수 있는 정보의 성질에 관계된다.

표 2-1에 암호분석자들에게 알려 진 정보량에 기초한 암호분석공격의 여러가지 형태를 종합하여 주었다. 가장 어려운 경우는 **암호문만 주어 지는 경우**이다. 일부 경우에는 암호알고리듬조차 알려 지지 않지만 일반적으로는 적이 암호알고리듬을 알수 있다고 가정한 환경에서 가능한 한가지 공격은 모든 열쇠를 다 시험해 보는 맹목적인 공격수법이다. 만일 열쇠공간이 대단히 크다면 이 방법은 비현실적인것으로 된다. 즉 적은 암호문자체의 분석에만 의존해야 하며 일반적으로 각이한 통계적검사를 적용한다. 이러한 수법에 리용하기 위하여 적은 영어 혹은 프랑스어본문, MS-DOS EXE파일, Java원천목록, 음성파일 등과 같은 주어 진 평문의 형태에 대한 어떤 일반적지식을 가져야 한다.

표 2-1. 암호화된 통보문에 대한 공격들의 형태

공격의 형태	분석자가 알고 있는 내용
암호문	• 암호알고리듬 • 분석해야 할 암호문
기지평문	• 암호알고리듬 • 분석해야 할 암호문 • 비밀열쇠를 리용하여 만든 하나 혹은 몇개의 평문,암호문쌍
선택평문	• 암호알고리듬 • 분석해야 할 암호문 • 분석자가 선택한 평문과 그에 대응하는 암호문
선택암호문	• 암호알고리듬 • 분석해야 할 암호문 • 분석자에 의해 선택된 암호문과 그에 대응하는 평문
선택문	• 암호알고리듬 • 분석해야 할 암호문 • 분석자에 의해 선택된 평문과 대응하는 암호문 • 분석자에 의해 선택된 암호문과 비밀건에 의해 생성된 대응하는 해석된 평문

암호문공격은 적이 정보해석에 필요한 최소의 정보만 가지므로 해석을 방지하기가 매우 쉽다. 많은 경우에 해석자들은 더 많은 정보를 가진다. 해석자들은 어떤 암호문과 꼭 같은 평문통보문을 몇개 포착할수 있다. 또 어떤 해석자는 어떤 평문의 패턴이 어느 통보문에 나타날수 있는가를 알수 있다. 실례로 Postscript형식에서 부호화된 어떤 파일은 늘 같은 패턴으로 시작된다. 즉 거기에는 표준화된 머리부나 전자자금전송통보문의 표재 등이 포함되군 한다. 이러한것들은 기지평문의 실례이다. 그에 대한 지식을 가지고 해석자들은 알려 진 평문이 전송되는 방법에 기초하여 그 열쇠를 추론해 낼수 있다.

기지평문공격에 밀접히 관계되는것은 확률단어공격과 같은것이라고 볼수 있다. 만일 적이 일반문제의 통보문에 대한 암호에 대하여 분석하고 있으면 그는 무엇이 그 통보문에 있는가를 다소 알수 있다. 더우기 적이 특별히 지정된 정보를 가지게 되면 통보문의 그 부분은 알수 있을것이다. 실례로 어떤 전체 회계표들이 전송된다면 적은 그 파일의 머리부에서 어떤 열쇠단어의 배치를 알수 있을것이다. 다른 실례로 프로그람개발회사 X에 의하여 개발된 어떤 프로그람에 대하여 원천부호는 어떤 표준위치에 승인문(허가)을 포함할것이다.

만일 해석자가 어떤 방법으로 선택한 통보문을 그 체계에 삽입하는 원천지체계를 얻을수 있다면 선택평문공격이 가능하다. 이러한 전략의 한 실례는 3장에서 취급한 각이한 암호분석이다. 그 열쇠의 구조를 알아 내기 위해 기대할수 있는 패턴을 심중히 골라 낼수 있다.

표 2－1에 서로 다른 두가지 공격형태를 보여 주었다. 즉 선택암호문공격과 선택평문공격이다. 이것들은 분석기술로서는 일반적으로 적게 리용되나 가능한 공격방법들이다. 상대적으로 약한 알고리듬은 암호문공격에 견디지 못한다. 일반적으로 암호알고리듬은

기지평문공격에 견디게끔 설계된다.

두가지 정의를 더 지적할수 있다. 암호방식은 그 방식에 의하여 생성된 아무리 많은 암호문을 리용하여도 특정한 암호문에 대응하는 평문을 유일하게 결정할수 있는 충분한 정보를 포함하지 않으면 **무조건안전**이라고 한다. 즉 적이 아무리 많은 시간을 들여도 그 암호문을 분석하는것은 불가능하다. 1회매몰과 같은 방식을 제외하면 무조건 안전인 알고리듬은 없다. 암호알고리듬의 사용자들은 보통 다음의 두 기준 혹은 그중 하나를 만족하는 알고리듬을 추구하고 있다.

- 암호를 파괴하는 비용이 그 정보의 비용을 넘는다.
- 암호를 파괴하는데 드는 시간이 그 정보의 실제적수명보다 더 길다.

암호방식이 우의 두 기준에 부합되면 **계산량적으로 안전**이라고 말한다. 난점은 암호문을 성공적으로 분석하는데 드는 품을 평가하기 어렵다는것이다.

표 2-2. 완전열쇠탐색에 필요한 시간

열쇠의 크기	가능한 열쇠들의 개수	1번의 암호화에 요구되는 시간, ㎲	10^6번의 암호에 요구되는 시간, ㎲
32	2^{32}=4.3×10^9	2^{31}㎲=35.8min	2.15ms
56	2^{56}=7.2×10^{16}	2^{55}㎲=1142년	10.01h
128	2^{128}=3.4×0^{38}	2^{127}㎲=5.4×10^{24}년	5.4×10^{18}년
26자모 치환	26!=4×10^{26}	2×10^{26}㎲=6.4×10^{12}년	6.4×10^6년

암호문으로부터 평문에로의 의미 있는 변환이 얻어 질 때까지 모든 가능한 열쇠를 가지고 시험하는 힘내기공격에서 요구되는 시간을 고찰할수 있다. 평균적으로 모든 가능한 열쇠의 절반만 시험해 보면 성공할수 있다. 표 2-2에 각이한 열쇠공간에 대하여 얼마만한 시간이 걸리는가를 보여 주었다. 3개의 2진열쇠들에 대해 고찰하였다. 56bit크기의 열쇠가 DES알고리듬에서 쓰인다. 결과는 또한 26개 문자열쇠를 리용하는 모든 가능한 치환(후에 론의함)의 경우로 보여 주었다.

매개 열쇠의 크기에 대하여 한번의 분석을 진행하는데 1㎲가 걸린다고 가정하였다(이 가정은 현대 콤퓨터에서는 타당하다). 극소형처리기의 병렬조작에 의하여 계산속도를 훨씬 개선할수 있다. 표 2-2의 마지막렬에서 매 ㎲당 10^6번의 시험을 할수 있다는 가정에서 결과를 주었다. 표를 통하여 알수 있는것처럼 이러한 성능수준에서 DES는 더는 계산량적으로 안전하다고 말할수 없다.

전통암호방식에 대한 모든 형태의 암호분석은 평문의 구조 또는 패턴의 흔적이 암호화에서 남아 있고 암호문에서 간파할수 있다는 사실을 리용하도록 설계되였다. 이것은 이 장에서 여러가지 전통암호방식들을 고찰하면 더 명백해 질것이다. 6장에서 공개열쇠암호방식에 대한 해석은 근본적으로 서로 다른 전제밑에서 전형화된다는것을 보게 될것이다. 즉 열쇠쌍의 수학적특성들이 두개 열쇠중 하나가 다른것으로부터 추출되게 할수 있다는것이다.

2.2 전자투과

엄밀히 말하여 암호가 아닌 전자투과(Steganography)에 대한 론의로부터 시작한다.

평문통보문은 두가지 방법중의 어느 하나로 은폐할수 있다. 전자투과방법은 그 통보문의 존재를 감춘다. 반면에 암호는 그 본문에 각이한 변환을 주어 3자에게 리해할수 없는 통보문을 준다. 표 2－3은 문헌 [KAHN96]에서 취하였는데 여러가지 보안방법에서 전자투과의 위치를 가리킨다. 칸(Kahn)에 의하면 **보안(Security)**은 정보를 보호하는 방법이고 **첩보활동(Intelligence)**은 정보를 검색하는 방법이다.

표 2－3. 암호보안과 첩보활동기술

암호보안	첩보활동
통신보안	통신첩보활동
전자투과(은현잉크, 공개부호들, 공개문속에 숨긴 통보문 등) 암호 (부호들과 암호) 거래보안 (호출수표변경, 허위통보문송신)	가로채기 암호분석 거래해석(통보문흐름분석, 무선지문뜨기)
전자적보안	전자적인 첩보활동
봉사보안(주파수변위, 스펙트르확산) 역방지대책(교란된 전파탐지를 통하여 보기)	전자정찰 (도청) 방지대책(전파탐지기통신방해, 거짓라지오신호의 발신)

전자투과의 단순한 형(그러나 구성에 시간이 소모된다.)은 명백한 의미가 없는 문자렬안의 문자나 단어렬에서 실제적인 통보문을 꺼내는것이다.

실례로 전체 통보문의 매개 단어의 첫 문자들의 렬로 감추어 진 통보문을 구성할수 있다. 그림 2－3에 전체 통보문의 단어들의 부분모임으로 숨겨 진 통보문을 나르는데 리용되는 례를 보여 주었다.

그 외에도 지난 시기에 리용되여 온 기술들의 실례들은 다음과 같다.

- **기호표식:** 인쇄 혹은 타자된 본문의 선택된 문자들이 연필로 겹쓰이였다. 표식들은 그 종이가 밝은 빛과 어떤 각도를 유지하지 않으면 보통 볼수 없다.
- **은현잉크:** 많은 물질들이 그 종이에 화학적작용 혹은 열이 가해 지기 전에는 그 어떤 가시적흔적도 남기지 않는다.
- **핀구멍뚫기:** 선택된 문자우의 작은 핀 구멍들은 그 종이가 빛면에 마주 서지 않으면 보통 보이지 않는다.
- **타자기교정리봉:** 검은 리봉으로 타자한 행들사이에 리용되면 교정테프를 가지고 타자한 결과들은 다만 강한 빛아래에서만 볼수 있다.

비록 이 기술들이 낡은것처럼 보이지만 현재도 의의를 가진다. 문헌[WAYN93]에는 CD우의 프레임의 제일 아래자리비트들을 리용하여 통보문을 은폐하는 방법이 소개되였다. 실례로 Kodak photo CD의 최대분해능은 3072×2048화소이다. 그 매 화소는 24bit의

RGB색정보를 포함한다. 매 24bit화소의 제일 아래자리비트(LSB)는 그 화상의 질에 큰 영향을 줌이 없이 변화시킬수 있다. 결과 한개의 수자속성사진에 2~3MB의 통보문을 은페할수 있다. 이러한 전자투과수법을 사용한 쏘프트웨어제품들이 많이 개발되였다[JOHN97].

전자투과는 암호화에 비하면 많은 결함을 가지고 있다. 그것은 정보의 상대적인 작은 렬비트들을 감추기 위해 많은 간접비용을 요구한다. 오히려 앞 절에서 제안한 방식들을 리용하는것이 더 효과적일수 있다.

Dear george,

Greetings to all Oxford.Many thanks for your
letter and for the summer examination package.
All Entry Forms and FeesForms should by Friday
for final despatch to the syndicate by Friday
20th or at the very latest, I'm told by the 21st.
Admin has improved here, though there's room
for improvement still; just give us all two or three
more years and we'll really show you! Please
don't let these wretched 16+proposals destroy
your basic 0 and A pattern. Certainly this
sort of change,if implemented immediately,
would bring chaos

Sincerely yours,

그림 2-3. 모르스경부가 손에 쥔 수수께끼

왜냐하면 체계가 일단 발견되면 그것은 실제로 쓸모 없게 되기때문이다. 이 문제도 역시 삽입법이 어떤 종류의 열쇠에 의존한다면 극복할수 있다(실례로 문제 2.3을 보라). 또한 어떤 통보문은 처음에 암호화되고 다음에 전자투과법을 리용하여 은페할수 있다.

전자투과의 우점은 비밀통신자체가 있다는것이 발견되지 말아야 할 대방들사이에 사용할수 있다는것이다. 암호화는 이 통신내용이 중요하다든가 비밀이라든가를 표시하든가 또는 송신자나 수신자가 은페해야 할 정보를 가진 사람이라는것을 확인한다.

2.3 전통암호기술

이 절에서는 전통암호기술이라고 부르는 방법들에 대한 실례를 고찰한다. 이 기술에 대한 학습에서는 오늘날의 전통암호에 대한 기본수법들과 또 예견하여야 할 암호공격의 형태를 실례로 보여 준다.

첫째로 모든 암호기술들중에서 2개의 기본기초블로크들인 대입과 치환을 연구한다. 끝으로 대입과 치환을 결합한 체계를 론의한다.

대입기술

대입기술은 평문의 문자들을 다른 문자나 수자, 기호로 교체하는것이다. 만일 평문을 비트렬로 본다면 대입은 평문비트패턴을 암호문비트패턴으로 교체하는것을 의미한다.

씨저암호

최초의 가장 단순한 대입암호는 씨저가 만든 암호였다. 씨저암호에서는 매 문자를 그 자모순에서 3자리뒤의 문자를 대입한다. 실례로

평문: meet me after the toga party
암호문: PHHW PH DIWHU WKH WRJD SDUWB

자모의 순서는 원순환으로 정해 진다. 따라서 Z의 다음에는 A가 놓이게 된다. 문자들의 가능한 대입은 다음과 같다.

평문: a b c d e f g h I j k l m n o p q r s t u v w x y z
암호문: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

만일 매 문자에 자모에서의 순서번호를 부과하면 (a=1, b=2 등) 알고리듬을 다음과 같이 표현할수 있다. 매 평문문자 P에 대하여 암호문문자 C를 대입한다.

$$C=E(p)=(p+3)\bmod(26)$$

밀기값은 임의의 크기로 정할수 있으며 따라서 일반적으로 씨저의 알고리듬은 다음과 같다.

$$C=E(p)=(p+k)\bmod(26)$$

여기서 K는 1～25사이의 값을 취한다. 복호알고리듬은 단순하다.

$$p=D(C)=(C-k)\bmod(26)$$

만일 주어 진 암호문이 시저의 암호문이라는것을 알면 힘내기공격은 쉽게 수행된다. 가능한 모든 25개의 열쇠를 시험해 보는것은 쉽다. 그림 2-4에서 실례암호문에 이 전략을 적용한 결과를 보여 주었다. 이 경우에 평문은 3번째 행이다(이 책에서 평문은 소문자로, 암호문은 대문자로, 열쇠값은 사체소문자로 약속한다).

이 문제의 중요한 특징들은 힘내기공격분석법을 리용할수 있는것이다.

1. 암호화알고리듬과 복호화알고리듬이 알려 져 있다.
2. 다만 25개의 열쇠만 시도한다.
3. 평문의 언어는 알려 져 있으며 쉽게 인식할수 있다.

KEY	PHHW	PHD	IWHU	WKH	WRJD	SDUWB
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrcp	rfc	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	ojbv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlq
12	dvvk	dv	rwkvi	kyv	kfxr	grikp
13	cuuj	cu	qvjuh	jxu	jewq	fqhjo
14	btti	bt	puitg	iwt	idvp	epgin
15	assh	as	othsf	hvs	hcuo	dofhm
16	zrrg	zr	nsgre	gur	gbtn	cnegl
17	yqqf	yq	mrfqd	ftq	fasm	bmdfk
18	xppe	xp	lqepc	esp	ezrl	alcej
19	wood	wo	kpdob	dro	dyqk	zkbdi
20	vnnc	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwo	ixizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	ry	fkyjw	ymj	ytlf	ufwyd
25	qiix	qx	ejxiv	xli	xske	tevxc

그림 2-4.씨저암호에 대한 힘내기공격분석

대부분의 망상황에서 알고리듬은 알려 져 있다고 가정할수 있다. 힘내기공격분석을 불가능하게 하는 일반적방법은 많은 열쇠를 가지는 알고리듬을 리용하는것이다. 실례로 DES알고리듬(3장에서 취급)은 56bit의 열쇠를 리용하는데 열쇠공간의 크기는 2^{56}, 즉 7 $\times 10^{16}$이상의 가능한 열쇠를 가진다.

세번째 특징 역시 중요하다. 만일 평문의 언어가 알려 지지 않았다면 평문의 출력은 명확히 알수 없을것이다. 더우기 입력이 어떤 방식으로 생략되거나 압축되면 식별을 더 어렵게 한다.

```
˜+Wµ"— Ω–O)≤4(∞‡‚ ë~Ω%ràu·¯í ◊¯Z-
Ú≠2Ò#Åæ∂ œ«q7¸Ωn·®3NÔÚ Œz'Y-ƒ∞í[±Û_ èΩ,<NO¬±«˘xã  Åä£èû3Å
x}ö§kºÂ
_yÍ ˆ∆É] ¸¤ J/˙iTê&ı 'c<uΩ-
 ÄD(G WÄC~y_ÏöÄW PÔı«ÎÜ†ç],¤¡˘Ì^üÑπ˘≈˘L˘9OgflO¯&Œ≤ ¬≤ ØÔ§˝:
 ˘Œ!SGqèvoˆ ú\¸S>h<-*6ø‡%x´˝|ñÓ#≈˜my‰˘≥ñP‹,ﬁ Áj  Å◊¿˝Zù-
Ω¨Õ¯6ŒŸ{% „ΩÊó ¸ï π+ÁÎ˙úO2çSÿ´O-
2ÄflßÌ /@ˆ"∏KªªpŒπ¸úé^´3∑¯ö˘ÕZÌ"Y¬ŸΩœY› Ω+eð/˙‹Kƒ¿*+˜"≤û˜
B ZøK˜Qßÿüƒ¸!ÒflÎzsS/]›ÈQ ü
```

그림 2-5. 압축문의 실례

그림 2-5에 ZIP알고리듬으로 압축된 어떤 본문파일의 일부를 실례로 주었다. 만일 이 파일이 단일대입암호(바로 26개이상의 자모기호들을 포함하도록 확장된)로 암호화되면 그 평문은 힘내기공격으로 분석되여도 그 식별이 불가능하다.

단일자모암호

씨저의 암호는 25개의 가능한 열쇠만을 가지고서는 안전하지 못하다. 임의의 방법으로의 대입을 허용하여 열쇠공간의 크기를 확장할수 있다.

씨저의 암호를 다시 보자.

평문: a b c d e f g h i j k l m n o p q r s t u v w x y z

암호문: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

가령 암호문행이 26개 자모의 임의의 치환이 될수 있다면 26! 즉 4×10^{26} 개이상의 가능한 열쇠들이 있게 된다. 이것은 DES의 열쇠공간보다 차수로 10이 더 크다(100억배의 계산량). 이것은 힘내기공격기술로는 분석이 불가능한것으로 된다.

그렇지만 다른 공격이 가능하다. 만일 암호분석자들이 그 평문의 성질을 알고 있다면(실례로 압축 안된 영문) 분석자는 그 언어의 규칙성을 밝혀 낼수 있다. 그러한 암호분석을 어떻게 하는가를 보여 주는 하나의 실례를 든다. 풀려는 암호문은 다음과 같다.

UZQSOVUOHXMOVVGDOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

첫 단계는 매 문자의 상대빈도를 계산하고 영문의 표준빈도분포와 비교한다(그림 2-6). 만일 통보문이 충분히 길다면 이 기술만으로 충분할수 있다. 그러나 이것은 짧은 통보문이기때문에 정확한 대조는 기대할수 없다. 이 실례의 경우에 문자들의 상대빈도는 다음과 같다.

P 13.33 Z 11.67 S 8.33 U 8.33 O 7.50 M 6.67	H 5.83 D 5.00 E 5.00 V 4.17 X 4.17	F 3.33 W 3.33 Q 2.50 T 2.50 A 1.67	B 1.67 G 1.67 Y 1.67 I 0.83 J 0.83	C 0.00 K 0.00 L 0.00 N 0.00 R 0.00

우의 표와 영문의 평문을 비교하여 보면 암호문의 P와 Z는 평문의 e와 t 라고 추정할수 있는데 꼭 정확한것은 아니다. 마찬가지로 S, U, O, M, H는 {r, n, i, o, a, s}에 대응될것 같고 또 A, B, G, Y, I, J는 {w, v, b, k, x, q, j, z}의 어느것과 서로 대응할것 같다.

이러한 관점에서 많은 대응방안들이 있게 된다. 이렇게 림시적인 부과를 하고 어떤 통보문이 합리적인 후보로 될수 있는가를 조사해 보게 된다. 보다 규칙적인 방법은 다른 규칙성을 더 찾아 내는것이다. 실례로 그 본문에서 어떤 단어들을 알수 있다고 하자. 또는 암호문자의 반복되는 렬들을 찾고 그것들의 평문을 추론해 내려고 시도할수 있다.

강력한 도구로서 두 문자결합의 빈도를 찾아 보는 방법이 있다. 가장 공통적인 2중음글자는 th다. 암호문에서 가장 공통적인 2중음글자는 ZW인데 그것은 세번 나타난다.

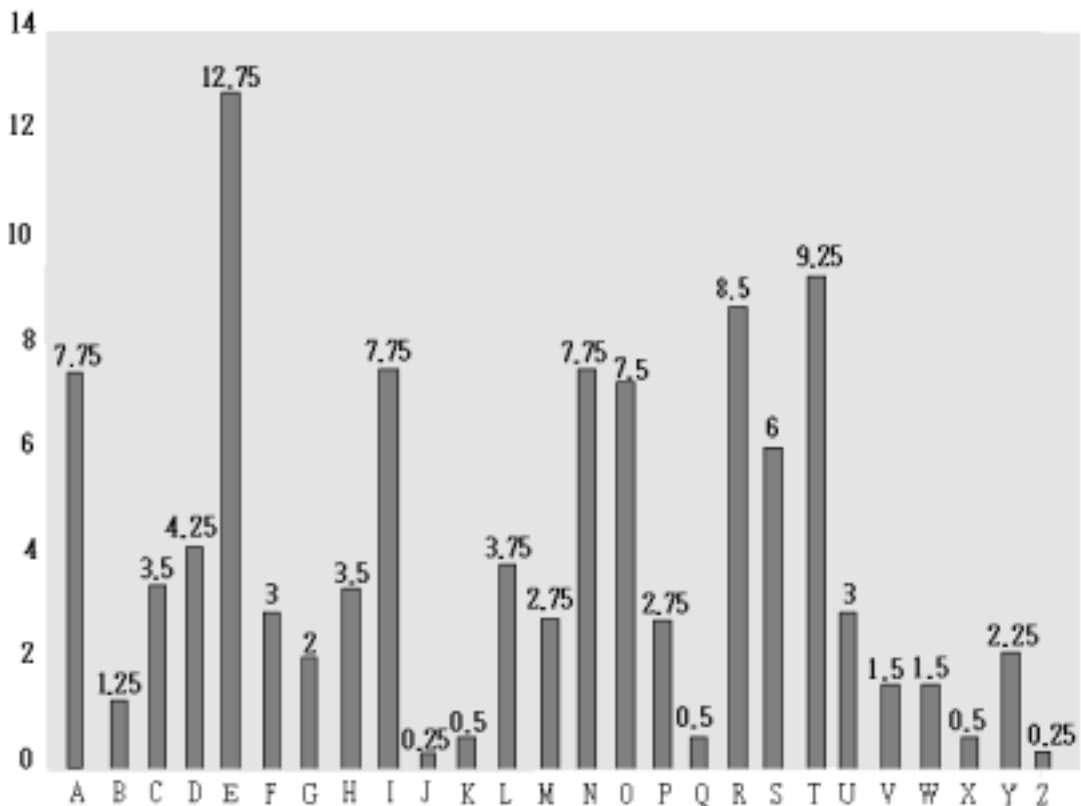



그림 2-6. 영어문장에서 글자들의 상대적빈도

따라서 Z는 t로, W는 h로 대응시킨다. 초기의 가정에 의하여 P와 e를 대응시킬수 있다. 그러면 암호문에서 “ZWP”가 나타나고 그것을 “the”로 넘길수 있다. 이것은 영어에서 가장 많이 나타나는 3문자결합이다.

다음 첫 행의 ZWSZ에 주의하자. 이 4개의 문자가 어떤 완전한 단어를 이루는지는 알수 없다. 그러나 만일 그렇다면 그것은 th-t로 될것이다. 따라서 s는 a와 같다. 이때 다음의 대응을 얻게 된다.

```
U Z Q  S U V U O H X  M O P V G P O Z P E V S G Z W S Z O P F
  t    a                  e     e   t e     a   t h a t   e

P E S X U  D B M E T S X A I Z V U E P H A H M D Z S H Z
e   a                a               e             a   t

O W S F P A P P D T S V P W U Z W Y M X U Z U H S X  E P Y E O H M Q
  h a   e   e e     a   e     t h         t     a      e
```

4개의 글자들만 확인되였지만 통보문의 일부분을 알게 되였다. 빈도와 시행착오의 결합의 계술적인 해석에 의하여 풀이를 쉽게 얻을수 있다. 단어들사이에 공백을 첨가한 완전한 평문은 다음과 같다.

> It was disclosed yesterday that sereral informal but
> direct contacts have been made with political
> representatives of the viet cong in muscow

단일자모암호는 그것이 원래 자모의 빈도자료를 반영하기때문에 쉽게 격파된다. 대책은 하나의 문자에 대하여 동음어로 알려 진 다중대입을 제공하는것이다. 실례로 문자 e에는 각이한 암호부호 16, 74, 35 혹은 21이 부과될수 있다(매 동음어들은 우연적 혹은 차례로 리용된다). 만일 매 문자에 부과되는 부호들의 개수가 그 문자의 상대빈도에 비례한다면 단일자모빈도정보는 완전히 무시된다. 유명한 수학자 칼 프리드리히 가우스(Carl Friedrich Gauss)는 동음어를 리용하여 파괴할수 없는 암호를 만들수 있다고 생각하였다. 그러나 동음어를 가지고서도 평문의 매 문자는 암호문의 하나의 원소에만 영향을 미치며 그 암호문에 대하여 비교적 정확하게 암호분석을 해도 다른 문자패턴(즉 빈도표)은 여전히 남아 있게 된다.

대입암호에서 평문의 구조가 암호문에 남을수 있는 범위를 줄이기 위하여 두가지 방법을 기본 리용하였다. 한가지 방법은 평문의 여러개의 문자들을 암호화하는것이고 다른 방법은 여러개의 암호자모들을 리용하는것이다. 그것들을 간단히 고찰하자.

Playfair암호

널리 알려 진 다중문자암호는 Playfair인데 그것은 평문에서 2중음글자를 하나의 단위처럼 처리하고 그것을 암호문의 2중음글자로 바꾼다(이 암호는 사실 영국과학자 챨리스 위트스톤(Sir charles wheatstone)이 1854년에 고안하였으나 그의 친우인 바론 플레이페어(Baron playfair)의 이름을 가지게 되였다. Playfair는 당시 영국외국인구락부의 암호권위자였다).

playfair알고리듬은 열쇠단어를 리용하여 만든 5×5문자행렬을 리용하는데 기초하였다. 그에 대한 하나의 실례가 있다.

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

이 경우에 열쇠단어는 monarchy이다. 행렬은 열쇠단어를 왼쪽으로부터 오른쪽으로, 우로부터 아래로 채우고 행렬의 나머지칸에는 열쇠단어에 속하지 않은 문자들을 자모순으로 채운다. 문자 I와 J는 한 문자처럼 취급한다. 평문은 다음의 규칙에 따라 한번에 두 문자로 암호화된다.

1. 같은 칸에 반복되여 들어 가는 평문문자들은 채움문자(실례로 X와 같은)로 분리시킨다. 그래서 balloon은 ba lx lo on와 같이 암호화된다.
2. 행렬의 같은 행에 떨어 지는 매개 문자들은 그의 오른쪽 문자로 바뀐다. 행의 원소는 순환적으로 마지막 원소에 련결된다. 즉 ar는 RM로 암호화된다.
3. 같은 렬에 놓이는 평문의 문자들은 그 아래의 문자로 바뀐다. 다만 렬의 맨 아래문자의 다음 문자를 그 렬의 첫 문자로 한다. 즉 mu는 CM로 암호화된다.
4. 기타 경우 매 평문문자는 행렬에서 그 두 문자를 정점으로 하는 직4각형의 다른 정점에 놓이는 문자로 바뀐다. 이때 평문의 문자에 대응하는 암호문의 문자는 대응하는 행에서 다른 정점에 놓이는 문자이다. 즉 hs는 BP로, ea는 IM(혹은 암호작성자의 의사에 따라 JM으로도)로 바뀐다.

playfair암호는 단순한 한자모암호에서의 큰 전진이다. 거기에는 26개 문자가 아니라 26×26=676개의 2중음글자가 있으므로 개별적인 2중음글자의 확인이 더 어렵다. 더우기 개별자모의 상대빈도는 2중음글자의 상대빈도보다 크게 변하며 따라서 빈도해석이 훨씬 어려워 진다. 이러한 리유로 하여 playfair암호는 오래동안 격파할수 없는것처럼 되여 왔다. 그것은 1차세계대전시기 영국군에서 표준적인 암호로 사용되였으며 2차세계대전시기에도 미군과 그의 동맹군에 의하여 많이 리용되였다. 그의 보안에서 기밀성수준이 그만큼 높았지만 이 암호는 비교적 쉽게 격파되였는데 그것은 거기에 본래의 평문의 많은 부분의 구조가 손도 대지 않은채로 남아 있었기때문이다. 대체로 암호문이 수백문자정도이면 충분하다.

playfair암호와 다른 암호의 효과성을 비교하는 한가지 방법을 그림 2-7(SIMM93)에 주었다. 평문을 의미하는 선은 암호학에 대한 Encyclopaedia Brittanica에서 7만개이상의 자모글자들 빈도분포를 주고 있다. 여기에는 또한 임의의 단일자모암호의 빈도분포를 주고 있다. 그림에서 점은 다음과 같은 방법으로 그렸다. 즉 본문에서 매 문자의 출현회수를 세고 문자 e의 출현회수로 나누었다(문자 e는 영어에서 빈도수가 제일 높다). 따라서 e는 1, t는 약 0.76 등이다. 수평축우의 점들은 빈도수가 줄어 드는 차례로 문자들을 배치한것이다.

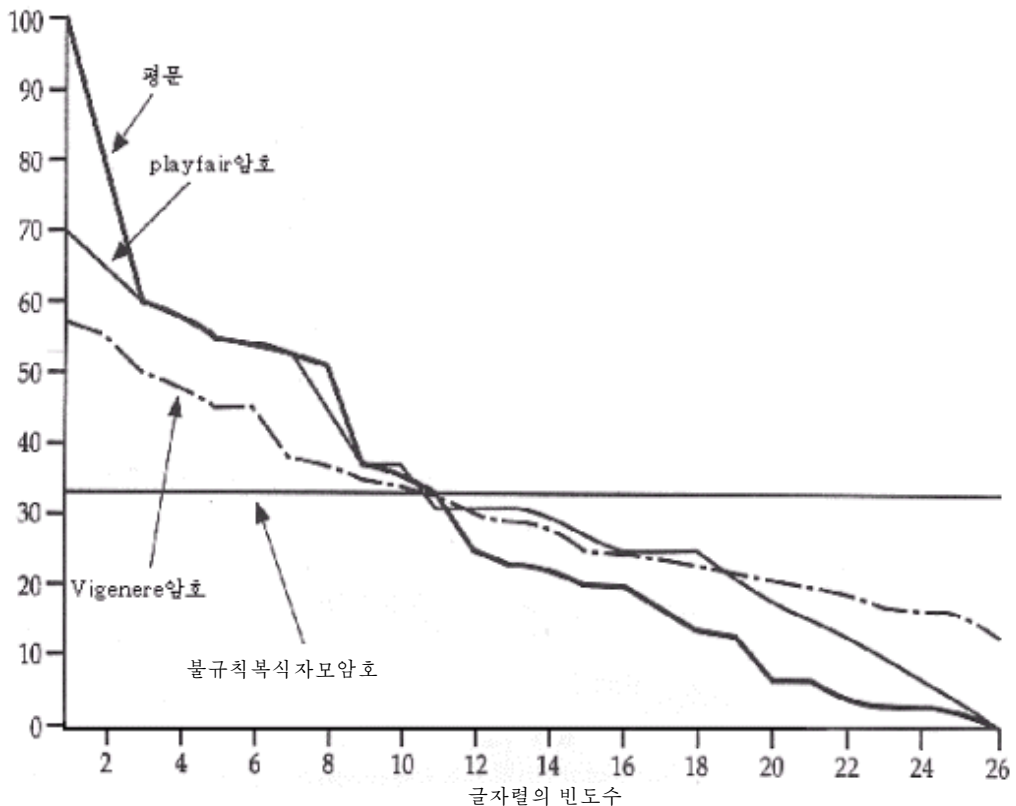



그림 2-7. 글자들의 상대적인 발생빈도수

또한 그림 2-7에서는 본문을 playfair암호로 암호화할 때 얻어 지는 빈도분포를 보여 주었다. 그라프를 정규화하기 위하여 암호문에서 매 문자의 출현회수를 평문에서 문자 e의 출현회수로 나누었다. 이렇게 얻어 진 그림은 문자들의 빈도분포가 암호화에 의해 은페되는 범위를 보여 준다. 만일 빈도분포정보가 암호화과정에 완전히 은페되였다면 암호문빈도곡선은 평탄할것이며 암호문에 대한 분석은 거의나 불가능할것이다. 그림에서 볼수 있는것처럼 playfair암호는 평문보다 더 평탄한 분포를 가지지만 그대신 암호해석자들이 처리해야 할 구조가 더 많아 진다.

Hill암호

흥미 있는 다중문자암호는 Hill암호인데 그것은 1929년에 수학자 레스터 힐(Lester Hill)에 의하여 제안되였다. 암호화알고리듬은 개개의 련속적인 평문문자에 대하여 m개의 암호문자들을 대입한다. 대입은 m개의 선형방정식에 의하여 결정되는데 거기서 매 문자에는 수값(a=0, b=1, …z = 25)이 부과된다. 실례로 m=3인 경우 체계는 다음과 같이 서술된다.

$$C_1 = (K_{11}P_1 + K_{12}P_2 + K_{13}P_3) \bmod 26$$
$$C_2 = (K_{21}P_1 + K_{22}P_2 + K_{23}P_3) \bmod 26$$
$$C_3 = (K_{31}P_1 + K_{32}P_2 + K_{33}P_3) \bmod 26$$

이것을 행렬과 벡토르로 표현할수 있다.

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix}$$

혹은

$$C = KP$$

여기서 C와 P는 길이가 3인 렬벡토르로서 각각 평문과 암호문을 표시하며 K는 3×3행렬로서 암호열쇠를 표현한다. 연산은 mod26으로 수행된다.

실례로 평문 "Paymoremoney"와 암호열쇠

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

를 리용하면 평문의 첫 세 문자는 벡토르 (15 0 24)로 표시된다. 이때 K(15 0 24)=(375 819 486) mod26=(11 13 18)=LNS이다. 이런 방식으로 계속하여 원래의 평문에 대응한 암호문 "LNSHDLEWMTRW"을 얻는다.

복호에는 K의 역행렬을 리용한다. K의 역행렬 K^{-1}은 식 $K^{-1}K=KK^{-1}=I$에 의하여 정의된다. 여기서 I는 단위행렬(주대각선에서만 1, 기타는 령)이다. 이 경우에

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

즉

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}\begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

K^{-1}을 암호문에 적용하면 평문이 얻어 진다. 선형대수와 관련한 내용에 대해서는 매우 간단히 취급하였는데 흥미 있는 독자들은 해당한 참고서를 리용할수 있다.

임의의 바른행렬 (m×m)에 대하여 행렬식은 매행, 매렬에서 꼭 하나의 원소를 취한 모든 적들과 어떤 항에 미누스부호를 붙여 취한 적들의 합과 같다(행렬식의 정의를 참고).

2×2행렬 즉

$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

에서 행렬식은 $K_{11}K_{22}-K_{12}K_{21}$이다. 3×3행렬에서 대하여 행렬식의 값은
$K_{11}K_{22}K_{33}+K_{21}K_{32}K_{13}+K_{31}K_{12}K_{23}-K_{31}K_{22}K_{13}-K_{21}K_{12}K_{33}-K_{11}K_{32}K_{23}$ 이다.

만일 바른행렬 A가 령 아닌 행렬식값을 가진다면 행렬의 거꿀행렬은 $[A^{-1}]_{ij}=(-1)^{i+j}(D_{ij})/\det(A)$이다. 여기서$(D_{ij})$는 A의 i번째 행과 j번째 렬을 제거하여 만든 부분행렬식의 값이고 det(A)는 A의 행렬식값이다. 계산에서 연산은 mod26에 의한것이다.

일반적으로 Hill암호체계는 다음과 같이 표현된다.

$$C=E_K(P)=KP$$
$$P=D_k(C)=K^{-1}C=K^{-1}KP=P$$

playfair와 마찬가지로 이 암호의 강도는 한문자빈도를 완전히 은페시키는데 있다. 사실 더 큰 행렬을 리용하면 더 많은 빈도정보를 은페시킬수 있다. 따라서 3×3 Hill암호는 단일자모빈도정보뿐아니라 두문자빈도정보도 은페한다.

Hill암호가 암호문공격에는 강하지만 기지평문공격(Known plaintext attack)에는 쉽게 격파된다. m×m Hill암호에 대하여 m개의 평문-암호문쌍을 가진다고 하자. 물론 매개의 길이는 m이다. 이 쌍들을 $P_j=(P_{1j},P_{2j},\cdots,P_{mj})$, $C_j=(C_{1j},C_{2j},\cdots,C_{mj})$ 으로 표시하면 $1\leqq j\leqq m$이고 몇개의 열쇠행렬 K가 알려 지지 않은 경우 $C_j=KP_j$이다. 두개의 m×m행렬을 정의하자. 그러면 행렬식을 $Y=XK$ 로 쓸수 있다. 만일 X 가 거꿀을 가진다면 $K=X^{-1}Y$ 이다. 만일 X 가 퇴화이라면(행렬식값이 령이라면) 퇴화 안되는 X 가 얻어 질때까지 평문-암호문쌍을 추가하면서 조작을 반복해야 한다.

문헌 [STIN95]에서 취급한 실례를 보겠다. 평문 "Friday"가 2×2Hill암호에 의하여 암호화되여 PQCFKU가 얻어 졌다고 하자. 즉 K(5 17)=(15 16); K(8 3)=(2 5); K(0 24)=(10 20)이다. 첫 두 평문-암호문쌍을 리용하면 다음과 같다.

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix}=\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}K$$

X의 거꿀은 다음과 같이 계산된다.

$$\begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1}=\begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}$$

따라서

$$K=\begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix}=\begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$$

이 결과는 나머지 평문－암호문쌍을 검사해 보면 명백히 확인된다.

다중자모암호

단순한 단일자모(monoalphabetic)기술에 대한 한가지 개선방법은 평문통보문을 처리할 때 각이한 단일자모대입을 리용하는것이다. 이 수법의 일반적이름이 다중자모암호이다. 이 모든 방법들은 다음과 같은 일반적인 특징을 가진다.

1. 해당 단일자모대입규칙들이 리용된다.
2. 열쇠는 어느 특정의 규칙이 주어 진 변환에 대해 선택되는가를 결정한다.

잘 알려 진 매우 단순한 알고리듬은 Vigenere암호이다. 이 방식에서 해당 단순자모대입규칙들은 26개의 씨저암호를 이룬다(0～25개의 밀기를 가지는). 매 암호는 열쇠글자에 의해 표시되는데 그것은 평문의 문자 a에 대하여 대입되는 암호문의 문자이다. 밀기 3을 가지는 씨저의 암호는 열쇠값 d로 표시된다.

방식의 리해와 리용을 돕기 위해 **Vigenere표**라고 부르는 행렬이 만들어 졌다(표 2－4). 26개의 매 암호문자는 수평으로 놓이며 그 표의 왼쪽에 매 암호에 대한 열쇠문자를 주었다. 평문에 대한 표준자모는 우에서부터 훑는다. 암호화과정은 다음과 같다. 가령 열쇠자모 X, 평문의 문자 y가 주어 지면 암호문자는 표의 X행 y렬에 놓이는 문자 V가 된다.

통보문을 암호화하는데는 그 통보문만큼 긴 열쇠가 필요하다. 흔히 열쇠는 열쇠단어들의 반복이다. 실례로 열쇠단어가 “deceptive”이고 통보문이 “We are discovered Save yourself”이면 그것의 부호화는 다음과 같다.

열쇠: deceptivedeceptivedeceptive
평문: wearediscoveredsaveyourself
암호문: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

복호화는 다음과 같다. 암호열쇠문자는 행을 결정한다. 암호문의 문자는 해당 행에서 렬을 결정한다. 복호된 문자(평문의 문자)는 정해 진 렬에서 꼭대기위치에 놓이는 문자이다.

이 암호의 강도는 매 평문문자나 열쇠단어의 매 문자에 대하여 다중암호문문자가 하나이라는데 있다. 그러므로 문자빈도정보는 은페된다. 그러나 평문구조에 대한 모든 정보가 잃어 지는것은 아니다. 실례로 그림 2－7에 길이가 9인 Vigenere암호에 대한 빈도분포를 제시하였다. playfair암호에 비해 상당히 개선되였으나 여전히 빈도정보가 남아있다.

이 암호를 격파하는 방법을 서술하는것은 교육학적과정과 관련된다. 그것은 이 방법이 암호분석이 적용되는 수학적원리에 기초하고 있기때문이다.

첫째로, 적은 암호문이 단일자모대입이나 Vegenere암호를 리용하여 암호화된것으로 알고 있다고 가정하자. 간단한 검사를 하여 그중 어느 하나인가를 결정할수 있다. 만일 단일자모대입이 리용된다면 그 암호문의 통계적성질은 대응하는 평문의 언어의 통계적성질과 같게 될것이다. 그래서 그림 2－6에서 알수 있는것처럼 어떤 암호문자는 상대적발생빈도수가 대략 12.75%이고 다른것은 대략 9.25% 등으로 된다. 만일 간단한 통보문만이 분석될수 있다면 이 표본통보문이 평문언어의 통계적거동과 정확히 일치할것이라고는 기대할수 없다. 그러나 그 대응이 가깝다면 어떤 단일자모대입을 가정할수 있다.

표 2-4.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

다른 한편 Vigenere암호가 짐작되면 처리는 인차 알수 있는것처럼 열쇠단어의 길이를 결정하는데 귀착된다. 먼저 열쇠단어길이를 어떻게 결정하는가를 보자. 요점은 다음과 같다. 만일 평문문자들의 동일한 렬들이 열쇠단어길이의 옹근수배수인 거리에서 나타난다면 그것은 일치하는 암호문문자렬을 발생시킬것이다. 우의 실례에서 두 기호렬 "red"사이의 거리는 9문자이다. 결국 두 경우에 r는 열쇠문자 e를 리용하여 또 e는 열쇠문자 P를 리용하여 그리고 d는 열쇠문자 t를 리용하여 암호화되였다. 그래서 두 경우에 암호문렬은 VTW이다.

암호문만을 주목하는 분석자들은 거리 9에서 반복되는 기호렬 VTW를 검출할수 있으며 그 열쇠단어는 길이가 3 혹은 9이라는 가정을 하게 된다. 그러나 그 통보문이 충분히 길면 거기에는 반복되는 암호문렬이 여러개 있을수 있다. 그 각이한 렬의 반복되는 거리의 공통인수를 고찰하여 분석자들은 열쇠단어의 길이에 대한 추측을 할수 있다.

이때 암호분석은 중요하게 통찰력에 의존한다. 만일 열쇠단어의 길이가 N이면 그 암호는 사실상 n개의 단일자모대입암호들로 구성된다. 실례로 열쇠단어 DECPTIVE에 의하여 위치 1, 10, 19 등에 있는 문자들은 모두 같은 단일자모암호로 암호화된다. 매개 단일자모암호를 제각기 공격하기 위하여 알려 진 평문언어의 빈도수특성을 리용할수 있다.

열쇠단어의 주기적성질은 통보문만큼 긴 비반복적인 열쇠단어를 리용함으로써 피할수 있다. Vigenere는 이른바 자동열쇠체계를 제안하였는데 거기서 열쇠단어는 열쇠단어에 실행열쇠를 제공하기 위하여 평문 그 자체와 련결된다. 이 경우에 다음과 같다.

열쇠:	deceptiveuearedisscoverdsar
평문:	wearediscoveredsavieyourself
암호문:	ZICVTWQNGKZEIIGASXSTSLVVWLA

이 방식도 암호분석에 약하다. 그것은 열쇠와 평문이 문자와 같은 문자빈도분포를 가지므로 통계적기술을 적용할수 있기때문이다. 실례로 그림 2-6으로부터 e에 의하여 암호화된 e는 $(0.1275)^2 \approx 0.0163$의 빈도로 발생하며 t에 의하여 암호화된 t는 이것의 약 절반으로서 발생한다. 이러한 규칙성은 암호분석자들이 암호분석을 실현하는데 리용될수 있다.

그러한 분석을 막기 위한 궁극적대책은 평문과 같은 길이를 가지며 그것과 아무런 통계적관련이 없는 열쇠단어를 선택하는것이다. 그러한 체계는 1910년에 힐베르트 버남(Gilbert Vernum)이라는 AT&T의 공학자에 의하여 제안되였다. 그 체계는 문자에서가 아니라 2진자료에서만 동작하였다. 체계는 다음과 같이 간단히 표현할수 있다.

$$C_i = p_i \oplus K_i$$

여기서

P_i = 평문의 i번째 2진수자
U_i = 열쇠의 i번째 2진수자
C_i = 암호문의 i번째 2진수자
$\oplus$ = 배타적론리합연산(or)

즉 암호문은 평문과 열쇠의 비트끼리의 XOR연산을 진행하여 생성된다. XOR의 성

질로부터 복호는 같은 비트렬에 대한 연산으로 실현된다. 즉

$$p_i = C_i \oplus K_i$$

이 기술의 본질은 열쇠를 만드는 방법이다. 버남(Vernam)은 그 열쇠를 실제로 반복시키는 테프의 실행순환을 리용할것을 제안하였는데 체계는 사실상 열쇠단어들이 반복되는 매우 긴 열쇠로 작업하였다. 그 방식에서 긴 열쇠를 써서 엄청난 암호분석곤난을 배출한다고 해도 그것은 충분한 암호문을 가지고 가능한 평문의 렬 또는 그것을 다 리용하여 파괴할수 있다.

죠세프 마으보그네(Joseph Mauborgne)는 보안의 한계에 도달한 버남암호의 개선을 제기하였다. 마드보그네는 반복되지 않는 통보문과 같은 길이의 우연적인 열쇠를 리용할것을 제안하였다. **한번쓰기받치개**로서 알려 진 이러한 방식은 격파할수 없다. 그것은 그 평문과 아무런 통계적관계도 가지지 않는 우연적인 출력을 내보낸다. 암호문은 평문에 대한 그 어떤 정보도 포함하지 않으므로 거기에는 그것을 해석하는 아무런 방법도 존재할수 없다. 이 방법을 실천적으로 적용하는데서 문제는 송신자와 수신자가 우연적이며 보호된 열쇠를 가지고 있어야 하는것이다. 따라서 그것은 독특한 암호임에도 불구하고 거의 리용되지 않는다.

전치기술

지금까지 론의된 모든 기술은 평문의 매 기호에 대한 암호문기호의 대입이였다. 매우 각이한 종류의 넘기기는 평문의 문자들에 대한 어떤 종류의 치환에 의하여 진행된다. 이 기술은 전치암호라고 부른다.

그러한 암호로서 가장 단순한것은 울타리보호기술인데 거기에서 평문은 대각선방향으로 씌여 지고 암호문은 행방향으로 읽어 낸다. 실례로 통보문 “meet me ofter the toga party”를 울타리보호기술로 암호화하기 위하여 다음과 같이 쓴다.

```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

암호화된 통보문은 다음과 같다.

MEMATRHTGPRYETEFETEOAAT

이러한것은 분석이 쉽다. 더 복잡한 방식은 직4각형으로 통보문을 가로 쓰고 렬의 순서를 치환하여 내리 읽어 내는것이다. 렬의 순서는 알고리듬에서 열쇠로 된다. 실례로

열쇠:	4	3	1	2	5	0	7
평문:	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	a	m	x	y	z

암호문: TTNAAPTMTSUOAODWCOIXKNIYPETZ

순수한 전치암호는 쉽게 분석할수 있다. 그것은 암호문이 초기평문과 같은 문자빈도를 가지기때문이다. 우의 렬전치형에서 암호분석은 간단한데 암호문을 행렬로 놓고 렬의 위치들에 대하여 분석을 진행한다. 2중음글자와 3중음글자의 빈도수표를 리용할수 있다.

전치암호는 전치를 여러 단계 진행하는것에 의하여 더 효과적인 보안을 보장한다. 결과 재구성하기 쉽지 않는 보다 복잡한 치환이 얻어 진다. 따라서 우의 통보문을 같은 알고리듬을 리용하여 재암호화하면 다음과 같다.

열쇠: 4 2 1 2 5 6 7
입력: t t n a a p t
m t s u o a o
d w c o i x k
n l y p e t z
출력: NSCYAUOPTTWLTMDNAOIEPAXTTOKZ

이 2중전치의 결과를 가시화하기 위하여 초기평문통보문의 문자를 그 위치에 부가된 수자로 표시한다. 따라서 통보문에서 28개 문자들에 대하여 원래의 문자들의 렬은 다음과 같다.

01 02 03 04 05 06 07 08 09 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28

첫 변환후에는 다음과 같다. 이것은 어느 정도 규칙적인 구조를 가진다.

03 10 17 24 04 11 18 25 02 09 16 23 01 08
15 22 05 12 19 26 06 13 20 27 07 14 21 28

2번째 변환후에는 다음과 같다.

17 09 05 27 24 16 12 07 10 02 22 20 03 25
15 13 04 23 19 14 11 01 26 21 18 08 06 28

이것은 훨씬 덜 구조화된 치환으로서 암호분석이 그만큼 더 힘들어 진다.

회전기계

우에서 고찰한 실례에서 알수 있는것처럼 암호분석을 더 어렵게 하는데서 다중암호화알고리듬이 효과적이다. 이것은 대입암호뿐아니라 전치암호에서도 마찬가지이다. DES의 도입이전에 암호화의 다중단계원리의 가장 중요한 응용은 회전기계라고 알려 진 체계들의 부류였다.

회전기계의 기본원리를 그림 2−8에 보여 주었다. 기계는 임플스전류가 흐르게 되여 있는 독립적으로 회전하는 원통들의 모임으로 이루어 졌다. 매 원통에는 각각 26개의 입력핀과 출력핀이 들어 있다. 내부배선은 매 입력핀에 유일한 출력핀을 련결시키도록 하였다. 간단히 하기 위하여 매 원통에서 3개의 내부련결만 제시하였다.

ㄱ) 초기설정 ㄴ) 한번 입력후 설정

그림 2—8. 번호판접촉에 의하여 표현된 배선을 가지는 3-회전기계

만일 매 입력핀과 출력핀을 자모의 문자와 맺어 준다면 하나의 원통은 어떤 단일자모대입을 정의한다. 실례로 그림 2—8에서 조작자가 문자 A에 대한 건을 누른다면 전기신호는 첫번째 원통의 첫 핀에 들어 가서 내부회로를 통하여 25번 출력핀에 간다.

한개의 원통을 가진 기계를 고찰하자. 매 입력건을 누른후에 원통은 한 위치만큼 회전한다. 따라서 내부결선도 한 자리 밀린다. 이렇게 하여 어떤 다른 단일자모대입암호가 정의된다. 평문의 26개 문자들다음에 원통은 초기위치에로 돌아 간다. 결국 주기 26인 여러자모대입알고리듬으로 된다.

단일원통체계는 보잘것 없으며 강한 암호분석공격을 감당할수 없다. 회전기계의 위력은 여러개 원통의 리용에 있다. 여기서 한 원통의 출력핀은 다음원통의 입력핀과 련결된다. 그림 2—8에 3—원통체계를 보여 주었다. 그림의 왼쪽 절반에서 조작자가 첫번째 핀(평문문자 a)에 즉 입력은 세개의 원통을 지나서 두번째 핀(암호문의 문자 B)에 나타나는 과정을 보여 주었다.

여러개 원통의 경우에 조작자로부터 가장 먼곳의 하나의 입력은 매번 건눌림할 때마다 하나의 핀위치를 회전한다. 그림 2—8의 오른쪽은 한번의 입력이 있은후의 체계의 거동을 보여 주었다. 바깥원통의 옹근회전동안에 중간원통도 한 핀의 자리만큼 회전한다. 또 중간원통의 옹근회전동안에 내부원통도 한 핀의 위치만큼 회전한다. 이것은 자동차의

거리기록계와 같은 원리로 동작한다. 결국 체계가 반복되기전에 26×26×26=17576개의 서로 다른 대입자모들이 있게 된다. 4개의 원통과 5개의 원통을 보충하며 각각 456976, 11011376개의 문자들의 주기가 얻어 진다. 다비드 카우(David Kahu)는 5회전기계에 대하여 다음의 내용을 지적하였다[KAHN96, 413페지].

그 길이의 주기는 문자빈도에 기초한 직접적인 풀이의 가능성을 좌절시킨다. 이 일반적 풀이방법에서 매 암호문자당 약 50개의 문자가 필요될것이다. 그 의미는 모든 5개의 회전기계들이 그 결합순환을 50번 지나야 한다는것이다. 암호문은 국회에서의 모든 연설들처럼 길 수 있다. 암호분석자들은 자기의 생존시에는 그러한 《진상품》을 가질것 같지 않다.

회전기계의 의의는 오늘 가장 널리 리용되고 있는 암호인 DES의 방법을 명시한데 있다. 이에 대하여 3장에서 구체적으로 보게 된다.

참고문헌

암호작성이나 파괴에 대하여 흥미를 가지는 독자들은 문헌[KAHN96]을 참고할수 있다. 그 문헌에서는 기술적개괄보다도 암호학의 방향에 대하여 더 집중하였다. 그러나 안내서로서 흥미 있는 자료로 될수 있다. 그밖에 암호기술을 취급한 단행본들이 많다[GARD72]. 많은 책들에서 보다 기술적인 측면에서 전통암호학을 취급하고 있다. 여기서 가장 중요한 문헌은 [SINK66]이다. 문헌[KORN96]에서는 친절하게 2진기술에 대하여 구체적이고도 친절하게 소개하였다. 문헌[SIMM93]에서는 명료성과 간결성이 독특하다. 그 책의 14페지에 암호학의 력사와 암호화기술에 대한 개괄이 심도 있게 소개되고 있다.

회전기계에 대한 구체적인 수학적취급은 문헌 [KOHM81]에 있다. 전자투파에 대한 해당 참고서는 문헌 [WAYN96]이다.

GARD72 Gardner, M. *Codes, Ciphers, and Secret Writing.* New York:Dover, 1972.
KAHN96 Kahn, D. *The Codebreakers: The Story of Secret Writing.* New York: Scribner, 1996.
KOHN81 Konheim, A. *Cryptography: A Primer.* New York: Wiley, 1981.
KOHN96 Korner, T. *The Pleasures of Counting.* Cambridge. England: Cambridge University Press, 1996
SIMM93 Simmons, G. "Cryptology." *Encyclopaedia Britannica*, 1993.
SIMK66 Sinkov, A. *Elementary Cryptanalysis: A Mathematical Approach.* Washington, DC: The Mathematical Association of America 1966.
WAYN96 Wayner, P. *Disappearing Cryptography*. Boston: AP Professional Books, 1996.

문 제

1. 그림 2-3에 들어 있는 통보문은 무엇인가?
2. 이 문제의 목적은 1회용에 대한 비파괴성을 보여 주는것이다. 27개의 자모에 의한 Vigenere방식을 리용한다고 가정하자(여기서 27번째 문자는 공백기호이다). 열쇠의 길이는 통보문의 길이와 같다. 암호문이 다음과 같이 주어 졌다고 하자.
   ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

다음의 평문을 주는 열쇠를 찾으시오.

MR MUSTARD WITH THE CANDLESTICK IN THE HALL

또한 다음의 평문을 주는 다른 열쇠를 찾으시오.

MISS SCARLET WITH THE KNIFE IN THE LIBRARY

결과를 설명하시오.

3. 도로씨 세이어(Dorothy Sayer)의 수수께끼들중에서 로드 페터(Lord Peter)는 그림 2－9에 제시된 통보문과 부닥친다. 그는 옹근수들의 렬인 통보문의 다음과 같은 열쇠도 찾는다.

   787656543432112343456567878878765654
   343211234345656787887876565443321123 4

   ㄱ) 통보문을 분석하시오. 암시: 가장 큰 옹근수값은 무엇인가?
   ㄴ) 알고리듬은 알지만 열쇠는 모른다면 그 방식은 얼마나 안전한가?
   ㄷ) 열쇠는 알지만 알고리듬을 모른다면 그 방식은 얼마나 안전한가?

4. 다음의 암호문은 단순한 대입알고리듬을 리용하여 생성되였다.

   53‡‡†305))6*;4826)4‡);806*;48†8¶60))85;;]8*;:‡*8†83
   (88)5*†;46(;88*96*?;8)*‡(;485);5*†2:*‡(;4956*2(5*−4(8¶8*
   ;4069285);)6†8)4‡‡;1(‡9;48081;8:8‡1;48†85;4)485†528806*81
   (‡9;48;(88;4‡?34;48)4‡;161;:188;‡?;

   이 통보문을 분석하시오. 암시:

   1) 영어에서 빈도가 가장 높은 문자가 e이라고 하면 통보문에서 빈도가 첫번째 또는 두번째인 문자를 e로 볼수 있다. e는 쌍으로도 나타날수 있다(실례로 meet, fleet, speed, seen, been, agree 등). 그 암호문에서 e로 복호되는 암호문의 문자를 찾으시오.
   2) 영어에서 가장 공통적으로 나타나는 단어는 "the"이다. 이 사실을 리용하여 t나 h로 복호될 기호를 찾으시오.
   3) 보조단어들을 추론하여 나머지 통보문을 복호하시오.

   주의: 결과통보문은 영어이나 처음 읽을 때 의미가 잘 안겨 오지 않을수 있다.

5. 열쇠배포문제를 푸는 하나의 방법은 송신자와 수신자가 다같이 가지고 있는 어떤 형식의 책을 리용하는것이다. 대표적으로 정탐소설에서는 책의 첫 문장이 열쇠로 될수 있다. 이 문제에서 론의되는 특수한 방식은 비밀부호를 포함하는 모호한 소설의 한 문장을 론의한다. 실례로 라드 렌델(Rath Rendell)의 소설 《Talking

to strange men》를 고찰하자. 이 문제를 그 책의 내용과 관계없이 고찰하자.

I thought to see the fairies in the fields, but I saw only the evil elephants with their black backs. Woe! how that sight awed me! The elves danced all around and about while I heard voices calling clearly. Ah! how I tried to see—throw off the ugly cloud—but no blind eye of a mortal was permitted to spy them. So then came minstrels, having gold trumpets, harps and drums. These played very loudly beside me, breaking that spell. So the dream vanished, whereat I thanked Heaven. I shed many tears before the thin moon rose up, frail and faint as a sickle of straw. Now though the Enchanter gnash his teeth vainly, yet shall he return as the Spring returns. Oh, wretched man! Hell gapes, Erebus now lies open. The mouths of Death wait on thy end.

그림 2-9. 로드 페터가 부닥친 수수께끼

다음과 같은 통보문이 있다.

SIDKHKDM AF HCRKIABIE SHIMC KD LFEAILA

이 암호문은 *The Other Side of Silencen*의 첫 문장을 리용하여 만들었다(첩자 Kim Philby에 대한 책).
단순대입암호문이 리용되였다.

ㄱ) 암호화알고리듬은 무엇인가?
ㄴ) 얼마나 안전한가?
ㄷ) 열쇠배포문제를 단순화하기 위하여 대방들은 열쇠로서 그 책의 첫 문장 또는 마지막 문장의 리용을 약속할수 있다. 열쇠를 변경하려면 어떤 새 책에 대한 합의가 필요하다. 첫 문장의 리용은 마지막 문장의 리용보다 더 적합하다. 왜 그런가?

6. 샬로크 홈스(Sherlock Holmes)는 다음의 통보문을 받았다고 하자.

534 C2 13 127 36 31 4 17 21 41
DOUGLAS 109 293 5 37 BIRLSTONE
26 BIRLSTONE 9 127 171

와트손(Watson)은 당황해 하였으나 홈스(Holms)는 암호의 형태를 곧 추출하였다. 당신은 어떻게 하겠는가?

7. 일반적인 단일자모암호의 약점은 송신자와 수신자가 다같이 기억기에 치환된 암호렬을 기억해 두어야 하는것이다. 이것을 피하는 일반적인 기술은 암호렬을 생성할수 있는 열쇠단어를 리용하는것이다. 실례로 열쇠단어 *CIPHER*를 리용하여 표준순서로 비상용문자들을 열쇠단어뒤에 써내고 그것을 평문의 문자들과 대조시킨다.

평문: a b c d e f g h i j k lm n o p q r s t u v w x y z
암호문: C I P H E R A B D F G J K L M N O Q S T R V W X Y Z

만일 이 파정이 충분한 혼합을 얻어 내지 못하게 되면 나머지 문자들을 아래와 같이 가로 쓰고 렬방향으로 읽어 낸다.

```
C I P H E R
A B D F G J
K L M N O Q
S T U V W X
Y Z
```

그러면 다음의 렬을 얻는다.

C A K S Y I B L T Z P D M U H F N V E G O W R J Q X

그러한 체계는 3절의 실례에서 리용하였다. 열쇠단어를 결정하시오.

8. playfair암호에서 가능한 열쇠는 얼마나 많은가? 풀이를 2의 근사제곱으로 표시하시오.
9. 만일 평문—암호문쌍이 충분히 제공된다면 Hill암호는 기지평문공격에 의하여 격파된다. 선택평문공격을 쓰면 Hill암호는 더 쉽게 풀릴것이다. 그러한 공격을 설명하시오.

제 3 장. 전통암호: 현대기술

이 장에서는 현대전통암호의 원리를 서술한다. 이를 위하여 가장 널리 보급된 전통암호알고리듬인 **DES**(Data Encryption Standard)에 대하여 론의한다. DES가 나온 이후에 수많은 전통암호알고리듬들이 제안되였지만 DES는 여전히 중요한 암호알고리듬으로 되고 있다. 더우기 DES의 면밀한 연구는 다른 전통암호알고리듬에서 리용된 원리들을 리해하는데 도움을 준다. 4장에서는 그밖의 중요한 전통암호알고리듬들을 취급한다.

RSA와 같은 공개열쇠암호방식에 비해 볼 때 DES의 구조는 매우 복잡하며 RSA처럼 간단히 설명할수 없다. 따라서 수동적인 암호화와 복호의 실험은 알고리듬세부들의 동작과정을 옳바로 리해할수 있게 한다. 교수경험은 이러한 간단한 교재의 학습이 DES에 대한 리해를 도모한다는것을 보여 준다.

단순DES에 대한 론의에 이어 이 장에서는 대칭블로크암호의 일반원리를 고찰하는데 그것은 이 책에서 취급하는 전통암호알고리듬의 류형이다. 다음으로 DES를 전반적으로 취급하며 계속하여 블로크암호설계의 일반적론의를 더 심화시킨다.

3.1 단순DES

단순DES는 안전한 암호알고리듬이라기보다 교육적성격을 가진다. 그것은 DES보다 훨씬 적은 파라메터를 가지나 DES와 류사한 성질과 구조를 가진다. DES는 싼타클라라종합대학(Santa Clara University)의 에드와르드 쇄퍼(Eduward Schaefer)에 의하여 개발되였다. 독자들은 실례들을 수동적으로 해보는 과정을 통하여 더 잘 리해할수 있을것이다.

개괄

그림 3－1에 단순DES의 총적구조를 보여 주었다. 앞으로 단순DES를 S－DES로 표기하기로 한다. S－DES암호알고리듬은 입력으로서 8bit의 평문블로크와 10bit열쇠를 가지며 출력으로 8bit암호문블로크를 내보낸다.S－DES복호알고리듬은 8bit암호문의 블로크와 같은 10bit열쇠를 리용하여 본래의 8bit평문블로크를 내보낸다.

암호화알고리듬은 5개의 기능 즉 초기치환 (IP), 치환과 대입연산자를 다 가지며 열쇠입력에 의존하는 복소함수 f_k, 자료의 두 절반부분을 서로 바꾸는 단순치환함수 SW, 함수 f_k, 초기치환의 역치환 IP^{-1}을 가진다. 2장에서 언급한것처럼 치환과 대입의 다단리용은 알고르듬을 복잡하게 하며 그것은 암호분석을 더 어렵게 한다.

함수 f_k는 입력으로서 암호알고리듬을 통과하는 자료뿐아니라 8bit열쇠도 가진다. 알고리듬은 2개의 8bit부분열쇠로 이루어 지는 16bit열쇠로 작업하게끔 설계되였는데 f_k의 매 출현에 대하여 하나의 8bit부분열쇠만 리용되게 되여 있다. 교대적으로 단일 8bit－열쇠가 리용되는데 알고리듬에서 같은 열쇠가 두번 리용된다. 그림 3－1에서 보여 주는것처럼 절충적으로 두개의 8bit－부분열쇠를 발생하는데 10bit열쇠를 리용한다. 이 경우에 열쇠는 먼저 치환된다(P10). 다음 밀기연산이 수행된다. 이 밀기연산의 출력은 첫 부분열쇠 K_1를 위한 8bit출력(P8)을 생성하는 치환함수를 통과한다. 또한 밀기연산의 출력은 다른 밀기연산과 P8의 다른 구체례에 들어 가 두번째 부분열쇠 K_2을 만든다.

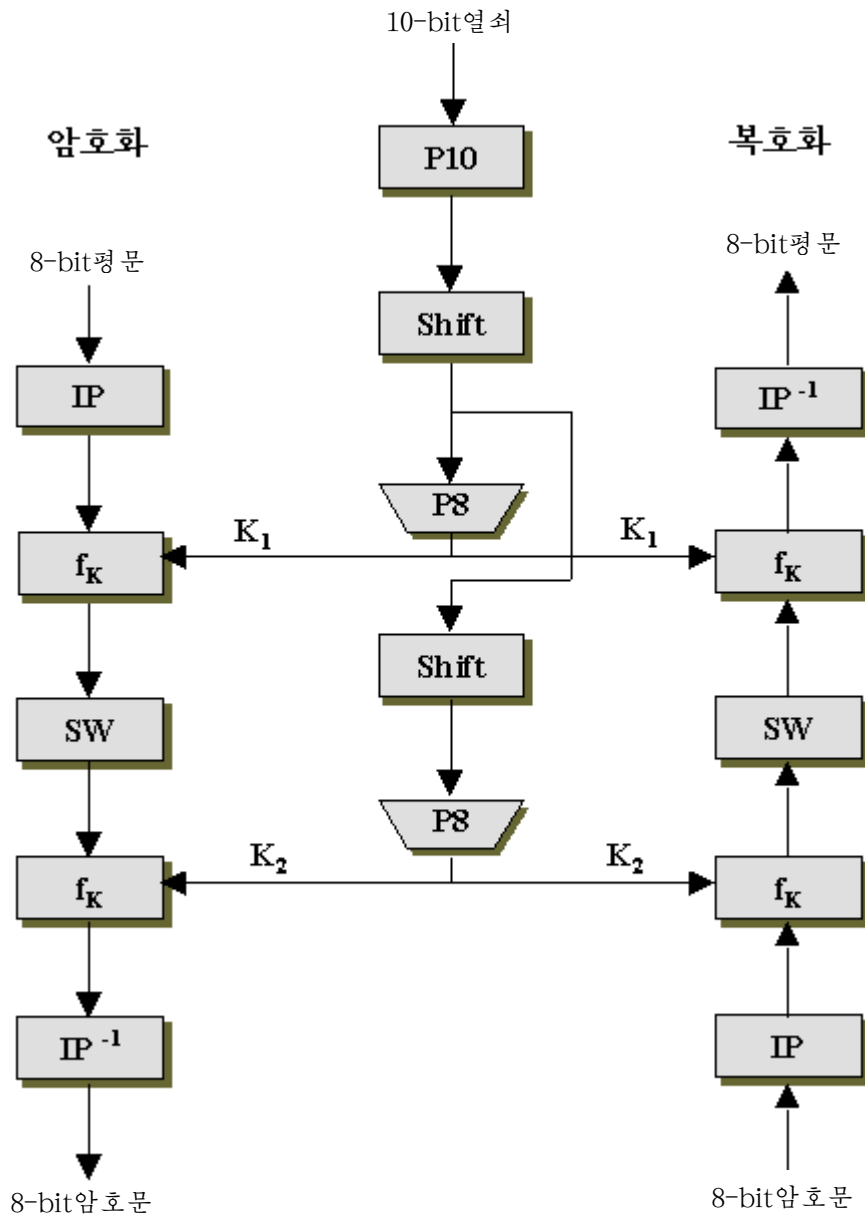



그림 3-1. S-DES도식

암호알고리듬을 함수의 합성으로 정확히 표현할수 있다. 즉

$$IP^{-1} \cdot f_{K_2} \cdot SW \cdot f_{K_1} \cdot IP$$

또는

$$\text{암호문} = IP^{-1}(f_{K2}(SW(f_{K1}(IP(\text{평문})))))$$

여기서

$$K_1=P8(Shift(P10(key)))$$

$$K_2=P8(Shift(Shift(P10(key))))$$

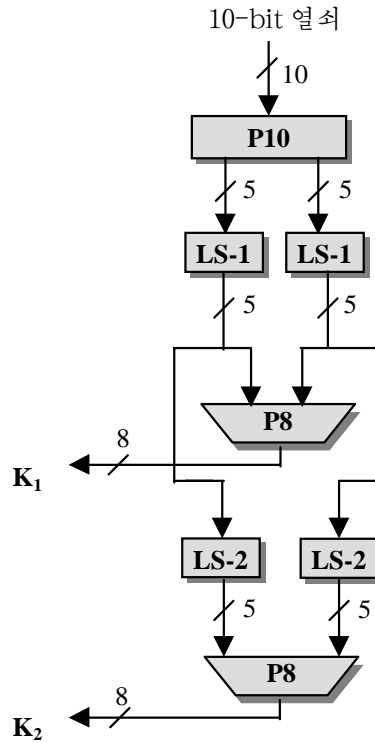



그림 3-2. S-DES에서의 열쇠생성

복호는 그림 3-1에 제시한것과 같으며 본질적으로는 암호화의 거꿀과정이다. 즉

$$평문=IP^{-1}(f_{K1}(SW(f_{K2}(IP(암호문)))))$$

이제 S-DES를 더 구체적으로 고찰하자.

S - DES 의 열쇠생성

S-DES는 송신자와 수신자가 공유한 10bit-열쇠의 리용에 의존한다. 이 열쇠로부터 두개의 8bit-부분열쇠를 만들어 암호화와 복호화알고리듬의 개별적단계들에서 리용한다. 그림 3-2에 부분열쇠들의 생성단계들을 보여 주고 있다.

다음과 같은 방식으로 열쇠를 취환한다. 먼저 10bit열쇠를 다음과 같이 표시하자 $(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10})$. 치환 P10를 다음과 같이 정의한다. 즉

$P10(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)$

이 P10을 그림으로 표시하면 다음과 같다.

P10									
3	5	2	7	4	10	1	9	8	6

이 표를 왼쪽으로부터 오른쪽으로 읽는다. 표에서 매 자리는 그 자리에서 출력비트를 내보내는 입력비트를 준다. 따라서 첫 출력비트는 입력의 3번째 비트, 두번째 출력은 입력의 5번째 비트 등과 같다. 실례로 열쇠(1010000010)은 열쇠(1000001100)로 치환된다. 다음에 결과를 수의 첫 다섯비트와 두번째 다섯비트에 대하여 각각 비트반전 혹은 한비트왼쪽순환밀기 (LS−1)을 수행한다. 우의 실례에서 결과는 (0000111000)으로 된다.

다음에 P8을 적용하는데 그것은 다음규칙에 따라 10bit중에서 8bit를 골라 내여 치환한다. 즉

P8							
6	3	7	4	8	5	1	9

이 결과가 부분열쇠 1(K_1)이다. 우의 실례에서 결과는 (10100100)이다.

다시 먼저 실시한 P10과 두개의 LS−1의 실행결과에 두비트순환밀기를 매 부분렬에 적용한다. 우의 실례에서 (0000111000)은 (0010000011)로 된다. 끝으로 P8을 다시 적용하여 K_2을 얻는다. 우의 실례에서 K_2은 (01000011)로 된다.

S−DES암호화

그림 3−3에 S−DES암호화알고리듬을 대략적으로 보여 주었다. 이미 언급한것처럼 암호화는 5개 함수들의 순차적인 적용과정을 포함한다. 이제부터 그 매 함수들에 고찰한다.

초기치환과 최종치환

알고리듬에 대한 입력은 평문의 8bit블로크인데 IP함수를 리용하여 먼저 치환한다.

IP							
2	6	3	1	4	8	5	7

이 과정에 평문의 8bit 모두가 그대로 유지되면서 다만 섞어 진다. 알고리듬의 마지막에 거꿀치환이 리용된다.두번째 치환이 실제로 첫 치환의 거꿀 즉 $IP^{-1}(IP(X))=X$이라는것을 실례를 통하여 쉽게 알수 있다.

IP^{-1}							
4	1	3	5	7	2	8	6

함수 f_k

S-DES에서 가장 복잡한 성분이 f_k인데 이것은 치환과 대입함수의 결합으로 이루어져 있다. 이 함수를 다음과 같이 표현할수 있다. 가령 L과 R를 각각 f_k에 대한 8bit입

력의 맨 왼쪽 4bit와 맨 오른쪽 4bit라고 하자. 그리고 F를 4bit렬로부터 4bit렬에로의 넘기기라고 하자(꼭 1:1넘기기일 필요는 없다).

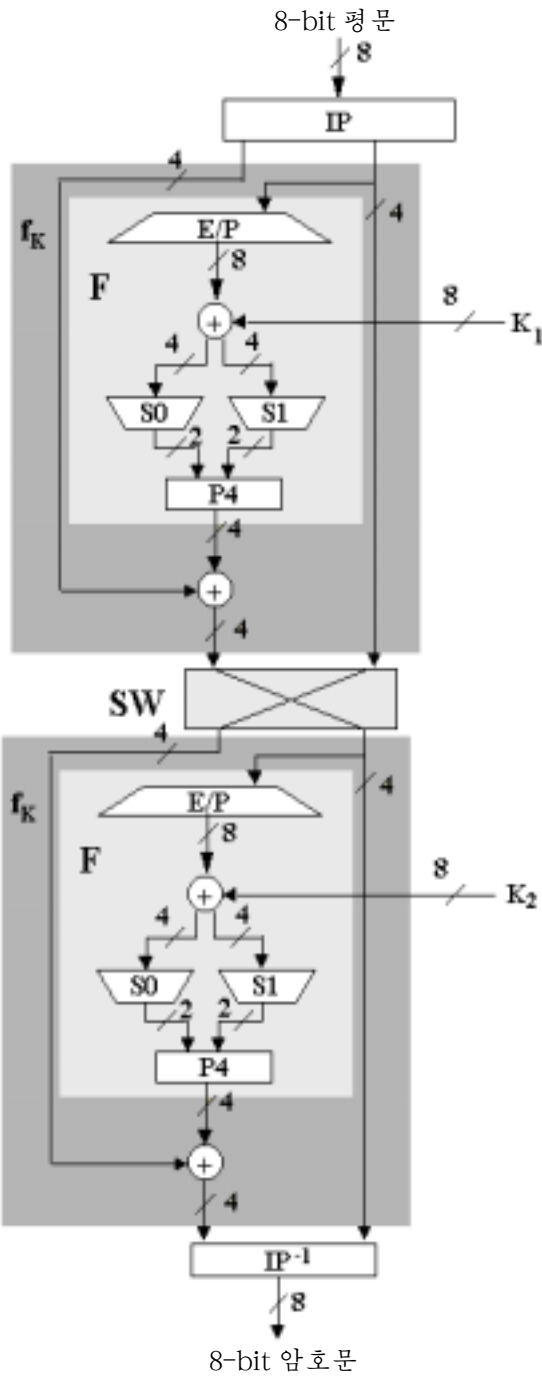



그림 3-3. S—DES방식의 암호화세부

그러면

$$f_k(L,R)=(L\oplus F(R,SK),R)$$

여기서 SK는 부분열쇠, $\oplus$는 비트별 배타적론리합이다.

그림 3－3에서 IP단계의 출력이 (10111101)이고 F(1101, SK)=(1110)(어떤 열쇠 SK)에 대하여)이면 f_k(10111101)=(01011101)이다. 여기서 (1011)$\oplus$(1110)=(0101)

넘기기 F를 보자. 이 넘기기의 입력은 4bit수(n_1 n_2 n_3 n_4)이고 첫 연산은 확대/치환연산이다.

E/P						
4	1	2	3	2	3	1

이것을 더 명백히 표현하면 다음과 같다.

$$\begin{array}{c|cc|c} n_4 & n_1 & n_2 & n_3 \\ n_2 & n_3 & n_4 & n_1 \end{array}$$

8bit－부분열쇠 $K_1=(k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{17}, k_{18})$은 이 값에 다음과 같이 배타적론리합된다.

$$\begin{array}{c|cc|c} n_4+k_{11} & n_1+k_{12} & n_2+k_{13} & n_3+k_{14} \\ n_2+k_{15} & n_3+k_{16} & n_4+k_{17} & n_1+k_{18} \end{array}$$

이 8bit들에 다시 이름을 달면 다음과 같다.

$$\begin{array}{c|cc|c} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} \\ P_{1,0ÿ} & P_{1,1} & P_{1,2} & P_{1,3} \end{array}$$

첫 4개 비트(우의 행렬의 첫행)는 S－통의 S0에 들어 가 2－bit출력을, 나머지 4bit(우의 행렬에서 두번째 렬)는 S_1에 들어 가 역시 2－bit출력을 생성한다. 이 두 통들은 다음과 같이 정의된다.

$$S_0=\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array} \qquad S_1=\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

S－통들은 다음과 같이 작용한다. 첫번째와 네번째 입력비트들은 S－통의 행을 지정하는 2bit수로서 처리된다. 또 두번째와 세번째 입력비트들은 S－통의 렬을 지정하는

2진수로서 처리된다. 그 행과 렬의 입력은 2를 기수로 하는 2－bit출력으로 된다. 실례로 $(P_{0,0}\ P_{0,3})$=(00), $(P_{0,1}\ P_{0,2})$=(10)이면 출력은 S0의 0행과 2렬로부터 나오는데 그것은 3 혹은 2진수로 (11)이다. 마찬가지로 $(P_{1,0}\ P_{1,3})$과 $(P_{1,1}\ P_{1,2})$는 S1에서 추가적인 2bit의 생성을 위하여 행, 렬의 첨수를 지정하는데 리용된다.

다음으로 S0과 S1에 의하여 생성된 4bit는 계속해서 다음과 같이 치환된다.

P4			
2	4	3	1

출력 P4는 함수 F의 출력이다.

절환함수

함수 f_k에 의해 입력의 맨 왼쪽 4개의 비트들만 변경된다. 절환함수 (SW)는 f_k의 두번째 부분이(나머지 절반비트렬) 다른 4개의 비트들에 작용하도록 오른쪽과 왼쪽의 4개의 비트들을 교체한다. 이 f_k의 두번째 부분에서 E/P, S0, S1와 P_4함수들은 앞에서와 같다. 열쇠입력은 K_2이다.

단순 DES 의 해석

단순DES에 대한 힘내기공격은 반드시 성공할수 있다.10bit열쇠에 대해서는 2^{10}=1024개의 가능한 경우가 있다. 암호문이 주어 지면 공격자는 매 가능한 경우를 시도해 보면서 그것이 옳은 평문인가를 결정하기 위해 그 결과를 해석한다.

암호분석이란 무엇인가? 가령 어떤 단순한 평문과 그에 대응하는 암호문을 알지만 암호열쇠는 모르는 기지평문공격(Known plaintext attack)을 고찰하자(여기서 평문은 (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8)로, 암호문은 (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8)로, 암호열쇠는 (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10})라고 기호약속한다). 그러면 매 c_i는 p_j와 k_j의 어떤 다항식 g_i로 결정된다. 따라서 암호알고리듬을 10개의 미지수를 가진 8개의 비선형방정식으로 표현할수 있다. 이 방정식에는 많은 가능한 풀이가 있게 되는데 그 매개를 계산하여 분석한다. 알고리듬에서 매 치환과 더하기들은 선형넘기기이다. 비선형성은 S－통에서 볼수 있다. 이 통들에 대한 방정식을 써보자. 리해를 돕기 위해 $(p_{0,0}\ p_{0,1}\ p_{0,2}\ p_{0,3})=(a,\ b,\ c,\ d)$와 $(p_{1,1}\ p_{1,2}\ p_{1,3}\ p_{1,4})=(w,\ x,\ y,\ z)$로 표시하고 4－bit출력은 ($q$, r, s, t)로 이름을 고쳐 달자. 그러면 S0의 연산은 다음 방정식으로 정의된다.

$$q = abcd + ab + ac + b + d$$
$$r = abcd + abd + ab + ac + ad + a + c + 1$$

여기서 모든 더하기는 mod 2이다. 이와 류사한 방정식들로 S1이 정의된다. 결국 선형넘기기들은 비선형넘기기들로 바꾸어 져 암호문비트들에 대한 복잡한 다항식들이 얻어 지는데 그로 하여 암호분석이 어렵게 된다. 문제의 규모를 리해하기 위해 2진산수에서 10개 미지수를 가진 다항식으로 된 방정식은 2^{10}개의 가능한 경우를 가질수 있다는데 주목하면 평균적으로 매 8개의 방정식들에 대하여 2^9개의 경우들을 생각하면 될것이다 (2^{10}중에서 절반은 성공).

흥미 있는 독자들은 기호처리프로그람으로 이 방정식을 찾을수 있다.

DES 와의 관계

DES는 64bit입력블로크에서 실행된다. 암호화도식은 다음과 같이 정의된다.

$$IP^{-1}\cdot f_{K_{16}}\cdot SW\cdot f_{K_{15}}SW\cdot SW\cdots f_{K_{1}}\cdot IP$$

16개의 48−bit부분암호열쇠를 계산하는데 56bit열쇠가 쓰인다. 56bit의 초기치환을 실시하고 그다음 48bit의 밀기와 치환이 반복적으로 적용된다.

암호화알고리듬에서 F는 4개의 비트(n_1, n_2, n_3, n_4)에 대하여 조작하는것이 아니라 32개의 비트(n_1, …, n_{32})에 대하여 조작한다. 초기확장/치환후에 48개의 비트출력은 다음과 같이 도식화될수 있다.

$$\begin{array}{c|cccc|c}
n_{32} & n_1 & n_2 & n_3 & n_4 & n_5 \\
n_4 & n_5 & n_6 & n_7 & n_8 & n_9 \\
\cdot & & \cdot & & & \cdot \\
\cdot & & \cdot & & & \cdot \\
\cdot & & \cdot & & & \cdot \\
n_{28} & n_{29} & n_{30} & n_{31} & n_{32} & n_1
\end{array}$$

이 행렬이 48bit부분열쇠에 배타적론리합된다. 여기서 8개의 행들은 8개의 S−통에 대응된다. 매 S−통은 4개의 행과 16개의 렬을 가진다. 웃행렬식에서 행의 처음과 마지막 비트들은 S−통의 어떤 행을, 중간의 4개의 비트들은 렬을 선택한다.

3.2 블로크암호원리

사실상 현재 쓰이고 있는 모든 대칭블로크암호알고리듬들은 페이스텔(Feistel)블로크암호라고 부르는 구조에 기초하고 있다. 때문에 페이스텔암호의 설계원리를 연구해 보는것이 중요하다. 그러므로 먼저 흐름암호와 블로크암호를 비교하고 다음에 페이스텔블로크암호구조에 대한 착상계기와 그 응용의 몇가지를 론의한다.

흐름암호와 블로크암호

흐름암호에서 수자자료흐름은 한번에 한 비트 혹은 한 바이트씩 암호화된다. 실례로 자동열쇠화된 비게네르(Vigenere)암호와 버남(Vernam)암호들은 고전흐름암호들이다. **블로크암호**는 평문의 어떤 한개의 블로크가 하나의 단위로 처리되여 같은 길이의 암호문을 만들어 내는 암호이다. 일반적으로 64bit의 블로크가 사용된다. 이 장의 뒤에서 설명되는 여러가지 연산방식을 리용하면 블로크암호는 임의의 흐름암호와 같은 효과를 나타낼수 있다.

블로크암호는 분석에 더 많은 품이 든다. 일반적으로 블로크암호는 흐름암호보다 응용범위가 넓다고 볼수 있다. 망에 기초한 전통암호로서 대체로 블로크암호를 리용한다.

따라서 이 책에서는 대칭암호에 대하여 블로크암호를 기본으로 론의한다.

페이스텔암호구조의 착상계기

블로크암호는 n bit의 평문블로크에 연산을 가하여 n bit의 암호문을 생성한다. 2^n개의 가능한 평문블로크를 생각할수 있는데 암호를 거꿀변환하자면(즉 복호하자면) 하나의 평문블로크로부터 꼭 하나의 암호문블로크가 생성되여야 한다. 즉 그 대응은 일대일이며 가역적이다. 이러한 변환을 가역적 혹은 비특이적(nonsingular)변환이라고 부른다. 다음의 실례는 n=2에 대하여 비특이변환과 특이(singular)변환을 보여 준다.

가역 넘기기

평문	암호문
00	11
01	10
10	00
11	01

비가역 넘기기

평문	암호문
00	11
01	10
10	01
11	01

후자의 경우에 01의 암호문은 두개의 평문블로크중 어느 하나에 의하여 생성될수 있다. 그러므로 가역넘기기로만 넘기기를 제한한다면 서로 다른 변환의 개수는 2^n!으로 될것이다.

그림 3－4에 n=4일 때의 일반적대입암호에 대한 론리를 보여 주었다. 4－bit입력은 16개의 가능한 입력상태들중의 하나로 되는데 그것은 대입암호에 대하여 16개의 가능한 출력상태중의 어느 하나로 유일하게 변환된다. 이때 입력비트들은 4개의 암호문비트에 의하여

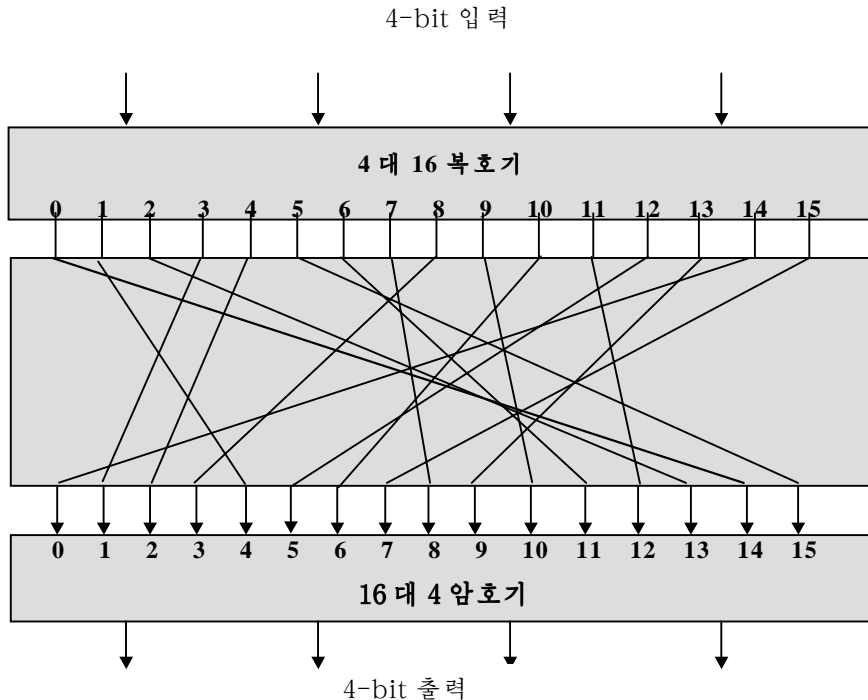



그림 3－4. 일반적인 n－bit－$n$bit블로크대입(n=4)

표시된다. 암호화와 복호화는 표 3-1에 보여 준것과 같은 도표작성에 의하여 정의할수 있다. 이것은 블로크암호의 가장 일반적인 형태이고 평문과 암호문사이의 임의의 가역넘기기를 정의하는데 리용될수 있다.

표 3-1. 그림 3-4의 대입암호에 대한 암호화 및 복호화표

평문	암호문
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

암호문	평문
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

그러나 여기에는 실천적으로 문제가 있다. 만일 블로크의 크기를 n=4와 같은 정도로 작은것을 리용하면 체계는 고전대입암호와 같아 진다. 그러한 체계들은 앞에서도 보았지만 평문의 통계적해석에 약하다. 이 약점은 대입암호의 리용에서 생기는 고유한것일뿐아니라 크기가 작은 블로크를 리용하는것으로부터도 생기는것이다. 만일 n이 크고 평문과 암호문사이에 임의의 가역적대입이 리용된다면 원천평문의 통계적특성은 이러한 암호분석이 불가능할 정도로 은페된다.

그러나 큰 블로크크기의 임의의 가역대입암호는 실행과 성능상 측면에서 볼 때 실천적이 못된다. 그러한 변환에서 넘기기되는것은 열쇠이다. 표 3-1을 다시 보자. 이 표는 n=4에 대하여 평문을 암호문에로 넘기는 하나의 특정의 가역넘기기를 보여 준다. 이 넘기기는 표의 두번째 렬의 내용(entry)에 의하여 정의되는데 그것은 매 평문블로크에 대한 대응하는 암호문블로크를 주고 있다. 이것은 본질에 있어서 모든 가능한 넘기기중에서 지정된 넘기기를 결정하는 열쇠이다. 이 경우에 열쇠는 64bit를 요구한다. 일반적으로 n bit의 대입암호문블로크에 대하여 열쇠의 크기는 $n\times 2^n$이다. 통계적공격을 막는데 필요되는 길이인 64-bit블로크에 대하여 열쇠의 크기는 $64\times 2^{64}=2^{70}\approx 10^{21}$bit이다.

이러한 문제점들을 고찰하면서 페이스텔은 큰 n에 대하여 쉽게 실현할수 있는 구성요소로 만들어 지는 리상적인 블로크암호체계에 근사시키는것이 중요하다고 지적하였다 [FEIS75]. 페이스텔의 방법을 취급하기전에 다른 하나의 사실에 주목하자. 론의를 일반적인 블로크대입암호로 국한시키고 쉽게 실현하기 위하여 2^n!개의 가능한 가역넘기기들중의 어떤 부분모임으로 고정한다. 실례로 선형방정식들의 어떤 모임으로 넘기기를 정의한다고 하자. N=4인 경우에

$$y_1 = k_{11}x_1 + k_{12}x_2 + k_{13}x_3 + k_{14}x_4$$
$$y_2 = k_{21}x_1 + k_{22}x_2 + k_{23}x_3 + k_{24}x_4$$
$$y_3 = k_{31}x_1 + k_{32}x_2 + k_{33}x_3 + k_{34}x_4$$
$$y_4 = k_{41}x_1 + k_{42}x_2 + k_{43}x_3 + k_{44}x_4$$

여기서 x_i는 평문블로크의 4개의 2진수자, y_i는 암호문블로크의 4개의 2진수자, k_{ij}는 2진결수, 연산은 mod 2이다. 열쇠의 크기는 n^2 즉 여기서는 16이다. 이러한 방법에서 부족점은 알고리듬의 구조를 아는 공격자가 암호를 분석할수 있다는것이다. 이 실례에서 언급한것은 본질상 2장에서 론의한 Hill암호인데 기호대신 2진수를 사용했을 따름이다. 2장에서 언급한것처럼 이와 같은 단순선형체계는 공격에 매우 약하다.

페이스텔암호

페이스텔은 적암호(product cipher)의 개념을 리용하여 단순대입암호를 근사시킬수 있다고 제안하였는데 그 방법은 마지막결과 혹은 생성물이 어떤 다른 암호성분보다 암호학적으로 훨씬 강하도록 둘 또는 그 이상의 기초암호들을 차례차례 실행하는것이다. 특히 페이스텔은 치환과 대입을 반복하는 암호의 리용을 제안하였다. 사실 이것은 혼란과 확산함수를 반복하는 적암호를 개발하기 위한 샤논(Shannon)의 제안의 실천적응용이다. 꼭 지적해야 할것은 페이스텔의 암호는 그것이 나온 때로부터 25년전(1945년) 샤논의 제안에 기초한것으로서 현재 가장 널리 쓰이는 대칭블로크암호라는것이다.

확산과 혼란

확산과 혼란이라는 말은 암호체계의 두개의 기초블로크로서 샤논이 도입한것이다. 샤논의 주되는 관심은 통계적해석에 기초한 암호분석을 막는것이였다. 그 추론은 다음과 같다. 가령 공격자가 평문의 통계적특성들에 대한 지식을 가지고 있다고 하자. 실례로 어떤 언어로 인간이 읽을수 있는 통보문에서 여러 문자들의 빈도분포를 알수 있다. 또는 단어들이나 그들의 빈도분포도 알수 있다. 만일 그 통계적성질이 어떤 방법으로 암호문에 반영된다면 암호분석자는 암호열쇠를 추론해 낼수 있거나 열쇠의 어떤 부분 또는 정확한 열쇠를 포함하는 적어도 어떤 모임을 추측해 낼수 있다. 샤논은 리상적으로 강한 암호란 암호문의 모든 통계적성질들이 리용되는 특정의 열쇠와 독립일것이라고 지적하였다. 먼저 론의한 대입암호의 경우인데 그러한것은 비실천적이라는것을 이미 보았다.

리상적인 체계에 의거하는것과는 달리 샤논은 통계적해석을 불가능하게 하는 두가지 방법인 혼란과 확산을 제안하였다. 확산에서 평문의 통계적구조는 암호문의 넓은 범위의 통계에로 분산된다. 이것은 매 평문의 수자가 많은 암호문수자값에 영향을 주게 하므로서 이룩되는데 그것은 매 암호문수자가 많은 평문수자들의 영향을 받는다는것과 같다. 확산의 실례는 어떤 통보문 $M = m_1, m_2, m_3, \cdots$을 어떤 평균화조작으로 다음과 같이 암호화하는것이다.

$$y_n = \sum_{i=1}^{k} m_{n+i} \pmod{26}$$

즉 하나의 암호문문자 y_n을 얻기 위하여 련속되는 k개 문자들을 더한다. 평문의 통계적구조가 확산된다는것은 분명하다. 따라서 암호문에서 문자의 빈도수들은 평문에서보

다 더 근사해 지며 2음절어빈도수들 역시 평문에서 보다 더 근사해 질것이다. 2진블로크암호에서 확산은 자료에 어떤 치환을 반복적으로 적용하여 실현되는데 이때 매 치환에는 어떤 함수의 적용이 따르는데 그에 의해 초기평문에서 각이한 자리의 비트들이 암호문의 한 비트를 결정하는데 참여하게 한다.

모든 블로크암호는 평문의 한 블로크를 암호문의 한 블로크로 변환하는것을 포함하는데 그 변환은 열쇠에 의존한다. 확산기구는 평문과 암호문의 통계적관계를 열쇠에 대한 추론을 어렵게끔 될수록 복잡하게 만든다고 볼수 있다. 한편 혼란은 암호문의 통계값들과 암호열쇠값사이의 관계가 그 열쇠를 발견해 내려는 시도가 좌절되도록 될수록 복잡하게 만든다. 즉 공격자가 암호문에서 어떤 통계값들을 얻어도 어떤 열쇠가 그 암호문을 만드는데 쓰이였는가를 추론해 내기 어렵게끔 복잡하게 하는것이다. 이것은 복잡한 대입알고리듬을 리용하여 이룩된다. 반면에 단순한 선형대입함수는 혼란에 거의 영향을 주지 못한다.

문헌[ROBS95]에서 지적한것처럼 현대블로크암호에서 성공의 여부는 확산과 혼란을 어떻게 조성하는가 하는것이다.

페이스텔암호구조

그림 3－5에는 페이스텔이 제안한 구조를 보여 주었다. 암호화알고리듬에 대한 입력은 길이 2*w* bit의 평문블로크와 열쇠 K이다. 평문블로크는 2개의 부분 L_0과 R_0으로 갈라 진다. 이 두 절반짜리 자료들은 *n*개의 처리단계를 거친 다음 결합되여 암호문블로크를 생성한다. 매 단계 i는 입력 L_{i-1}와 R_{i-1}를 가지며(이것들은 앞단계에서 얻어 진다.) 마찬가지로 열쇠 K에서 얻어 지는 부분열쇠 K_i를 가진다. 일반적으로 부분열쇠 K_i는 K와 다르며 K_i들도 서로 다르다.

모든 단계들은 같은 구조를 가진다. 대입은 자료의 왼쪽절반에 대하여 수행된다. 이것은 자료의 오른쪽절반에 단계함수 F를 적용하여 진행하며 다음에 그 함수의 출력과 자료의 왼쪽절반의 배타적론리합을 취한다. 단계함수는 매 단계에서 같은 일반적구조를 가지며 단계의 부분열쇠 K_i에 의하여 파라메터화된다. 이 대입 다음에는 그 자료의 두 절반들의 교체로 이루어 지는 치환이 실행된다. 이 구조는 샤논이 제안한 대입치환망(SPN)의 특수한 형태이다.

페이스텔망의 정확한 실현은 다음의 파라메터와 설계특성들의 선택에 의존한다.

- **블로크의 크기:** 블로크가 크면 클수록 안전성은 더 높아 진다. 그러나 암호/복호속도는 떠진다. 64bit의 블로크크기가 적당하며 거의 모든 블로크암호설계에서 리용한다.
- **열쇠의 크기:** 열쇠의 크기가 클수록 안전성은 높아 진다. 그러나 암호/복호속도를 줄인다. 64bit크기의 열쇠는 현재는 적합치 않으며 128bit가 공통적인것으로 되였다.
- **단계의 수:** 페이스텔암호의 본질은 한 단계는 보안에 불합리하지만 여러 단계는 안전성을 높인다는것이다. 일반적으로 16단계가 리용된다.
- **부분열쇠생성알고리듬:** 이 알고리듬이 복잡할수록 암호분석을 더 어렵게 한다.
- **단계함수:** 큰 복잡성은 일반적으로 암호분석에 더 잘 견디게 한다.

페이스텔암호의 설계에서 고려해야 할 다른 2가지 문제가 있다. 그 문제들은 이 장의 뒤부분과 다음 장들에서 론의하게 된다.

- **고속쏘프트웨어암호화 및 복호화:** 많은 경우에 암호화는 하드웨어실현을 배제하는 방식에서 응용 혹은 편의도구(Utility)기능에 구현된다. 따라서 알고리듬의 실행속도가 문제로 된다.
- **해석의 쉬움:** 암호분석이 될수록 어렵도록 알고리듬을 만들려고 해도 알고리듬에는 분석을 쉽게 하는 측면들이 적지 않다. 즉 알고리듬이 간결하고 명백히 설명될수 있다면 그것은 암호분석적약점으로 하여 그 알고리듬을 해석하기 쉽게 된다. 따라서 더 높은 수준의 강도를 가지도록 개발한다. 실례로 DES는 쉽게 분석되지 않는다.

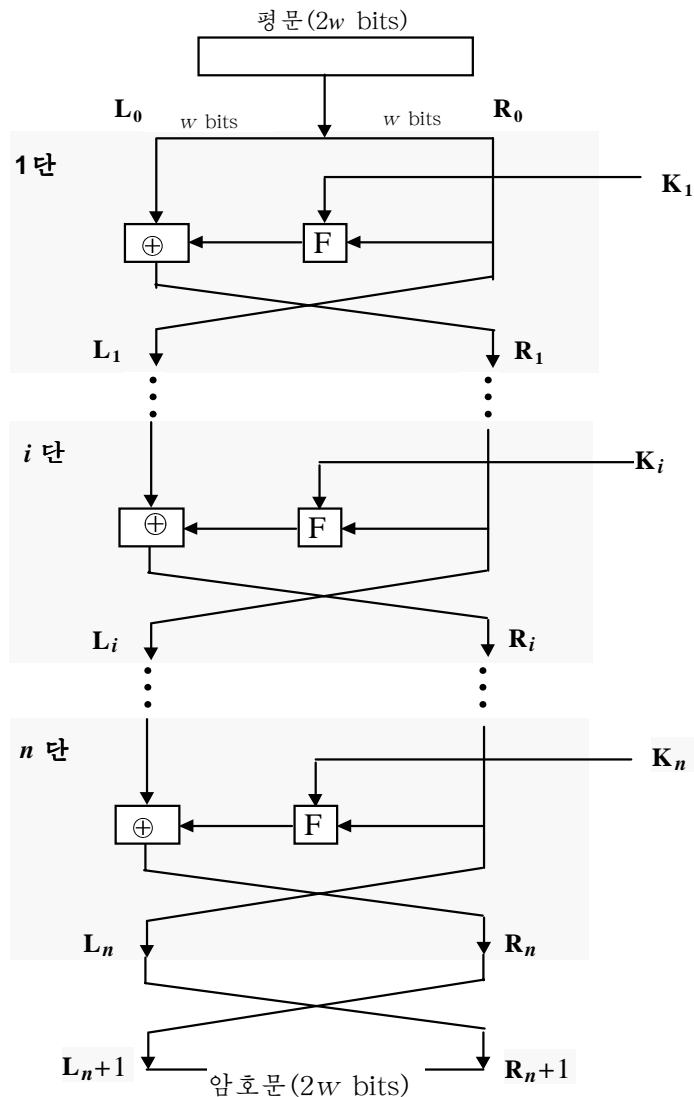



그림 3−5. 고전페이스텔망

그림 3—1과 그림 3—3을 다시 보면 SDES는 2단계의 페이스텔구조로 볼수 있다. 순 페이스텔구조와의 한가지 차이는 그 알고리듬의 시작과 끝이 치환기능을 가진다는것이다. 이 차이는 옹근DES(완전한 DES)에서도 나타난다.

페이스텔복호알고리듬

페이스텔암호에서 복호과정은 본질상 암호화과정과 같다. 그 규칙은 다음과 같다: 알고리듬의 입력으로서 암호문을, 그러나 부분열쇠 k_i는 거꾸순서로 리용한다. 즉 k_n은 첫 단계, k_{n-1}은 두번째 단계 등과 같은 방법으로 k_1가 마지막단계에서 리용된다. 이때 암호화와 복호화에서 같은 알고리듬을 리용하므로 좋은 알고리듬이다.

같은 알고리듬에 열쇠를 반대순서로 주면 정확한 결과를 주는가를 보기 위하여 그림 3-6을 고찰하자. 여기서는 암호화과정을 왼쪽에, 복호화과정은 오른쪽에 주었다(16단알고리듬결과는 단수에 관계 없다). 명백히 하기 위해 LE_i와 RE_i는 암호알고리듬에서 흐르는 자료를, RE_i 와 LD_i는 복호알고리듬에서 흐르는 자료를 표기한다. 그림은 매 단계에서 복호화과정의 중간값은 교환되는 값의 두 절반값을 가지는 암호화과정의 대응하는 값과 같다는것을 보여 준다. 다른 방법으로 설명하기 위해 i번째 암호단계의 출력을 $LE_i||RE_i$(L_i는 R_i와 련결된다.)로 표시한다. 그러면 대응하는 (16-i)번째 복호화단계의 입력은 $RD_i||LD_i$로 된다.

그림 3—6에 의해 우의 주장의 정당성을 증명하자. 암호화처리의 마지막 반복다음에 출력의 두 절반은 교환되며 그래서 암호문은 $RE_{16}||LE_{16}$이 된다. 그 단계의 출력이 암호문으로 된다. 이제 그 암호문을 같은 알고리듬의 입력으로 취한다. 첫 단계에 대한 입력은 $RE_{16}||LE_{16}$인데 그것은 암호화과정의 16번째 단계에서 출력의 32bit교환과 같다.

이제 복호화의 첫 단계의 출력이 암호화과정의 16번째 단계의 입력의 32bit교환과 같다는것을 보자. 먼저 암호화에서는

$$LE_{16}=RE_{15}$$
$$RE_{16}=LE_{15}\oplus F(RE_{15},K_{16})$$

복호화에서는

$$\begin{aligned}LD_1&=RD_0=LE_{16}=RE_{15}\\RD_1&=LD_0\oplus F(RD_0,K_{16})\\&=RE_{16}\oplus F(RE_{15},K_{16})\\&=[LE_{15}\oplus F(RE_{15},K_{16})]\oplus F(RE_{15},K_{16})\end{aligned}$$

XOR의 성질은 다음과 같다.

$$[A\oplus B]\oplus C=A\oplus[B\oplus C]$$
$$D\oplus D=0$$
$$E\oplus 0=E$$

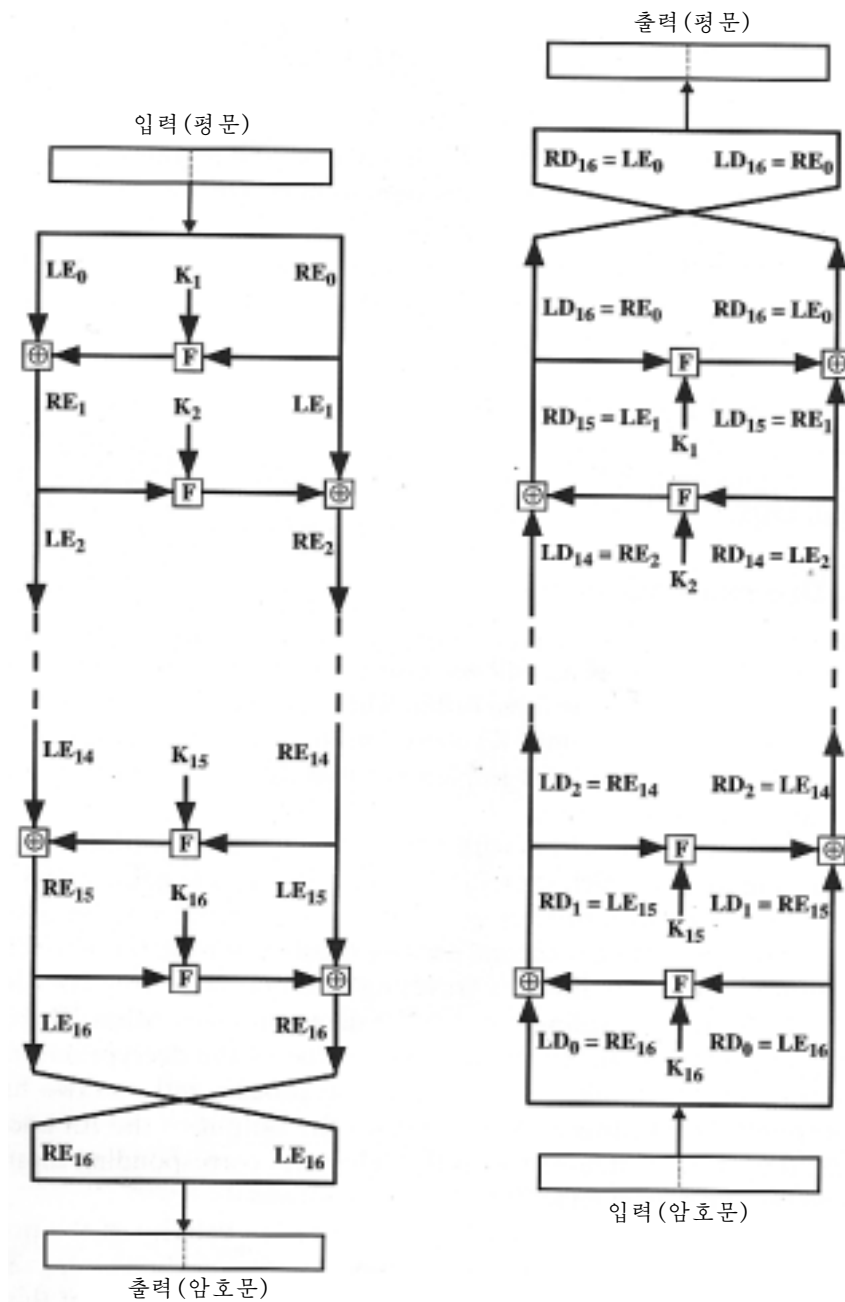



그림 3-6. 페이스텔의 암호화와 복호화

즉 $LD_1=RE_{15}$, $RD_1=LE_{15}$이다. 그로부터 복호화과정의 첫 단계의 출력은 $LE_{15}||RE_{15}$이다. 이것은 암호화의 16번째 단계의 입력의 32bit교환이다. 이 대응은 모든 16개의 단계에서 다 성립한다. 이것을 일반적으로 다음과 같이 표현할수 있다.

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$

다시 정돈하면

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus F(RE_{i-1}, K_i) = RE_i \oplus F(LE_i, K_i)$$

따라서 i번째 반복에 대한 입력을 그의 출력들의 함수로서 표시하였는데 이 등식들은 그림 3-6의 오른쪽에 보여 준 지정을 확증한다.

끝으로 복호화과정의 마지막단계의 출력이 $RE_0||LE_0$이라는것을 밝힌다. 32bit교환은 초기평문을 회복하며 따라서 페이스텔복호화과정의 정확성을 보여 준다.

이 론의에서 F가 가역함수일것을 요구하지 않는다. 그것을 보기 위하여 두 인수의 값에 관계없이 F가 상수출력을 내는 제한된 경우를 고찰하자. 이때도 등식은 성립한다.

3.3 자료암호화표준

가장 광범히 리용되는 DES암호도식은 1977년 NBS(지금의 NIST: 국가규격기술연구소)에 의하여 채택된 DES이다. DES에서 자료는 56bit의 열쇠를 리용하는 64bit블로크로 암호화된다. 알고리듬은 련속적으로 64bit입력을 64bit출력으로 변환한다.

광범히 리용되고 있는 DES에 대하여 그것이 얼마나 안전한가 하는 문제와 관련하여 많이 론의되고 있다. 론쟁의 성격을 파악하기 위해 DES의 력사를 간단히 고찰하자.

1960년대 말 IBM은 페이스텔에게 콤퓨터암호에 대한 연구과제를 맡겼다. 그 연구과정에 이른바 LUCIFCR라는 알고리듬이 개발되였다. 그것은 역시 IBM이 개설한 런던의 Lloyd회사의 현금처리체계에 리용되게 되였다. LUCIFER는 128－bit열쇠를 리용하여 64bit블로크로 조작하는 페이스텔블로크암호이다. LUCIFER연구에서 얻어 진 결과에 기초하여 IBM은 리상적으로 한 소편으로 실현할수 있는 상업용암호제품개발에 노력을 집중하였다. 그 연구를 왈터 투만(Walter Tuchman)과 칼 메이어(Carl Meyer)가 이끌었으며 IBM의 연구사들뿐아니라 외부의 전문가들과 NSA의 기술자들도 참가하였다. 결과 암호분석에 견딜수 있게 **LUCIFER**의 판본은 더 세련되였다. 다만 열쇠비트의 크기는 한 소편에 실장시키기 위해 56bit로 축소되였다.

1973년 NBS는 국가적인 암호규격에 대한 요구를 내놓았다. IBM은 투만－메이어의 연구결과를 제출하였다. 이것은 가장 훌륭한 알고리듬에 기초한것으로서 1977년 자료암호규격으로 제정되였다.

규격으로 제정되기전까지 제안된 DES는 맹렬한 비난을 받았으며 그것은 오늘까지도 가라앉지 않고 있다. 두 측면이 비평을 야기시켰다. 첫째로, 초기의 LUCIFER의 열쇠길이는 128bit였는데 제안된 체계는 다만 56bit로서 72bit라는 너무 많은 축소를 하였다는것이다. 비평가들에게는 열쇠의 길이가 너무 짧아 공격에 견딜수 있겠는가가 우려되였다. 둘째로, DES의 내부구조에 대한 설계기준이다. 즉 S－통문제이다. DES의 내부구조에 열쇠가 없이 통보문을 분석할수 있는 어떤 숨겨 진 약점은 없는가고 사용자들이 마음 놓을수 없었던것이다. 그후에 일어 난 사실들 특히 차분해석에 대한 최근의 연구는 DES가 매우 강한 내부구조를 가진다는것을 보여 주고 있다. 더우기 IBM관련자들의 말

에 의하면 제안에서 변화가 있었다면 NSA가 제기한 S—통들에서의 변화뿐인데 그 과정에 평가에서 제기되였던 약점들을 제거하였다.

DES는 특히 재정부분에서 많이 리용되였다. 1994년에 NIST는 앞으로도 DES의 리용을 재천명하였다. NIST는 DES가 비밀정보의 보호보다 응용프로그람들에서 더 광범히 쓸것을 권고하였다.

DES의 암호화

DES암호화에 대한 총적도식이 그림 3—7에 제시되였다. 임의의 암호방식과 마찬가지로 암호화하기 위하여 두개의 입력이 있게 된다. 즉 암호화하려는 평문과 열쇠이다. 여기서 암호문은 64bit길이로 블로크화되며 열쇠는 56bit이다.

그림의 왼쪽을 보면 평문의 처리는 세 단계로 진행됨을 알수 있다. 첫째로, 64bit평문은 허용되는 입력을 생성하기 위하여 초기치환된다(IP).

치환과 대입기능을 포함하는 동일한 16개의 단들로 이루어 지는 단계에 의하여 암호화가 진행된다. 마지막(16번째)단의 출력(암호기의 출력)은 입력평문과 열쇠의 함수인 64bit들로 이루어 진다. 출력의 왼쪽과 오른쪽 절반은 교환되여 예비출력을 생성한다. 끝으로 예비출력은 초기치환함수의 거꿀인 치환(IP^{-1})을 통과하여 64bit암호문으로 생성된다.

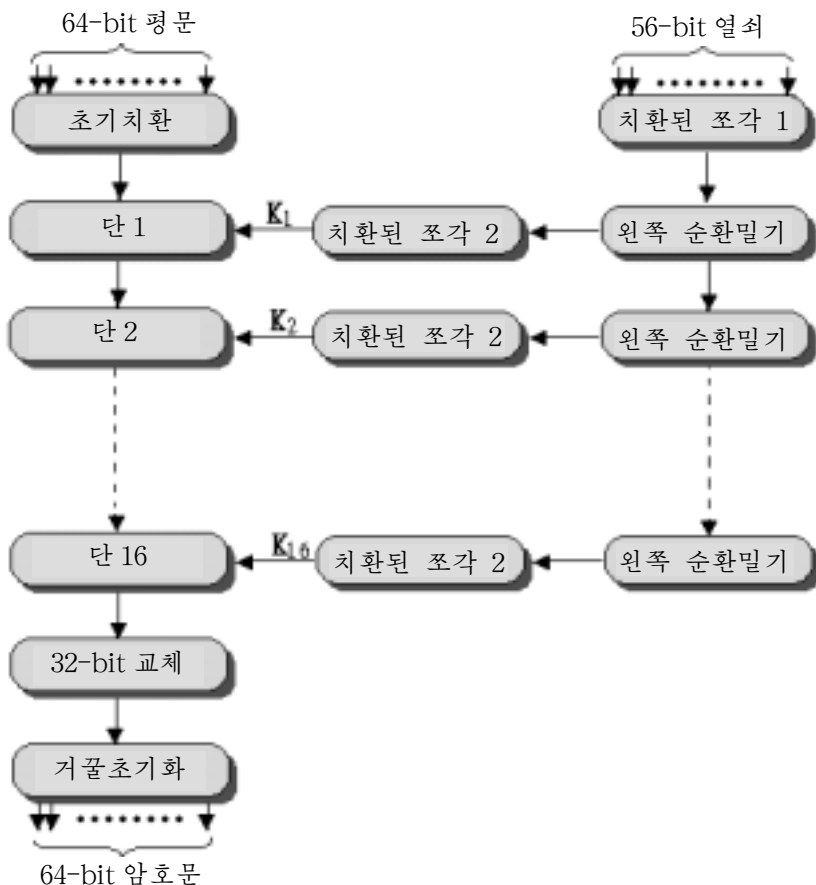



그림 3—7. DES암호화알고리듬의 일반적표시

초기치환과 최종치환을 제외하고 DES는 그림 3－5에 제시된것과 꼭같은 페이스텔암호구조를 가진다.

그림 3－7의 오른쪽 부분은 56bit열쇠가 어떻게 리용되는가를 보여 준다. 초기에 열쇠는 치환함수를 통하여 보내진다. 다음 16개의 매 단에서 부분열쇠 K_i는 왼쪽순환밀기와 치환의 결합에 의해 생성된다. 치환함수는 매 단에서 같지만 열쇠비트들의 반복에 의하여 서로 다른 부분열쇠가 생성된다.

표 3-2. DES의 치환표

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

ㄱ) 초기치환(IP)

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

ㄴ) 거꿀초기치환(IP^{-1})

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

ㄷ) 확장치환(E)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

ㄹ) 치환함수(P)

초기치환

초기치환과 그의 거꿀치환은 표 3-2에서와 같이 정의된다. 이 두 치환함수들이 실지 거꿀관계에 있는가를 보기 위하여 다음의 64－bit입력 M을 고찰하자.

M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

여기서 M_i는 2진수자이다. 치환 X=IP(M)은 다음과 같다.

M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_2
M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_4
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_6
M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_8
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_9	M_1
M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_3
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_5
M_{63}	M_{55}	M_{47}	M_{39}	M_{31}	M_{23}	M_{15}	M_7

만일 거꿀치환 $Y=IP^{-1}(x)=IP^{-1}(IP(M))$를 취하면 비트들의 초기배렬과 맞아 떨어 지는것을 알수 있다.

한개 단의 세부

그림 3-8에 한개 단의 내부구조를 보여 주었다. 그림의 왼쪽 절반을 보면 매 64bit 중간값의 왼쪽과 오른쪽 절반은 32bit덩어리(L(왼쪽), R(오른쪽))로서 각각 처리된다. 임의의 고전페이스텔2진암호에서와 마찬가지로 매 단에서의 총적처리는 다음의 도식으로 요약할수 있다.

$$L_i=R_{i-1}$$
$$R_i=L_{i-1}\oplus F(R_{i-1},\ K_i)$$

단열쇠 K_i는 48bit이다. 입력 R는 32bit이다. 이 입력 R는 R bit중에서 16bit의 복제를 포함하는 확장과 치환을 정의하는 표를 리용하여 48bit로 확장된다.

이때의 48bit를 K_i와 배타적론리합한다. 이 48bit의 결과는 표 3-2의 ㄹ에서 정의된것처럼 치환되여 32bit의 출력을 주는 대입함수를 통과한다.

함수 F에서 S-통들의 역할은 그림 3-9에 보여 주었다. 대입은 8개의 S-통들의 모임으로 이루어 지는데 그 매개는 6bit의 입력을 받아 4bit의 출력을 낸다. 이 변환을 표 3-3에

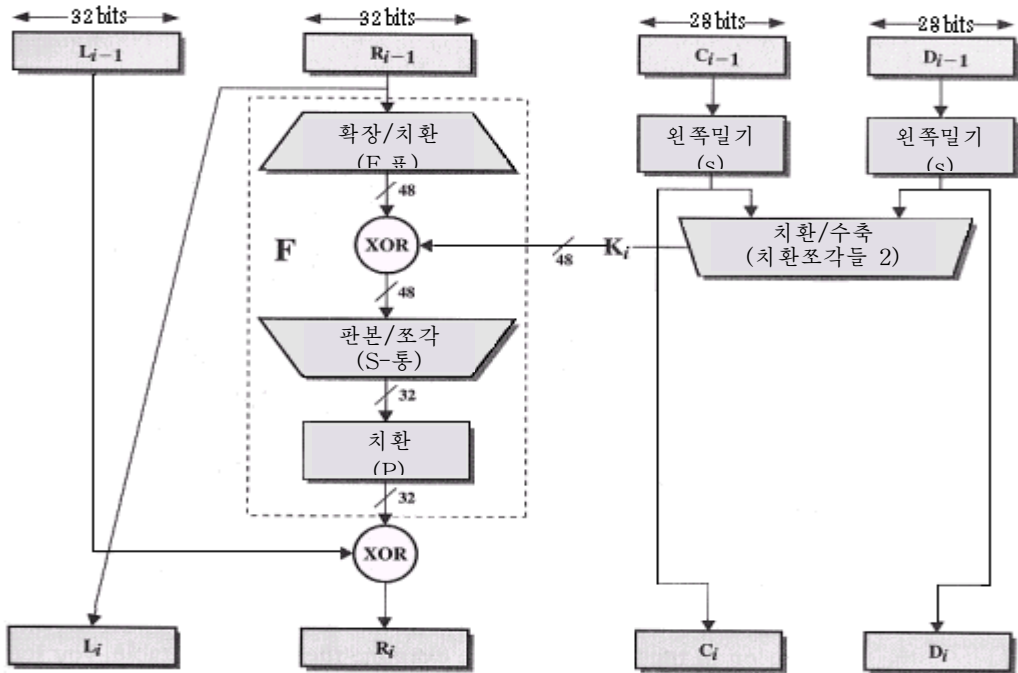



그림 3-8. DES알고리듬의 단

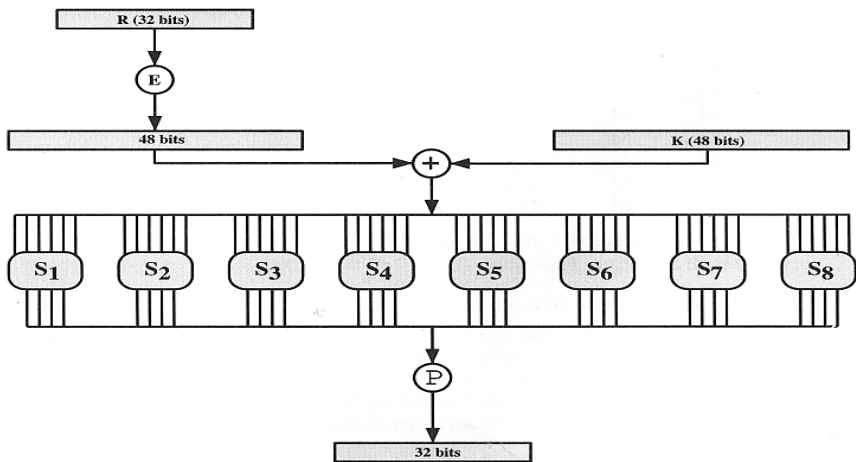



그림 3-9. F(R, K)의 계산

주었는데 그것을 다음과 같이 해석할수 있다. 통 S_i에 대한 입력의 처음과 마지막 비트들은 S_i에 대한 표에서 4개의 행에 의하여 정의되는 4개의 대입중에서 하나로 선택되는 2진수를 이룬다. 중간의 4bit들은 특정의 렬을 선택한다. 행과 렬에 의하여 선택된 세포의 10진값은 4bit 표현으로 변환되여 출력을 만든다. 실례로 S_1에서 입력이 011001일 때 행은 01 즉 1행이고 렬은 1100(12렬)이다. 1행 12렬에서 값은 9이다. 따라서 출력은

1001이다.

S—통의 매행은 일반적으로 가역대입을 정의한다. 그림 3—4를 통하여 넘기기를 리해할수 있다. 그림에서는 통 S_1의 행 0에 대한 대입을 보여 주었다.

표 3—3. DES S—통의 정의

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

S—통의 구조를 더 구체적으로 보면 다음과 같다. 열쇠 (K_i)의 의의를 잠시 무시하자. 확장표를 조사해 보면 입력의 32bit들은 4bit의 그룹들로 쪼개 지며 다음 2개의 린접한 그룹으로부터 바깥 비트들을 취하여 6bit의 그룹들로 된다는것을 알수 있다.

실례로 입력단어의 부분이

…efgh ijkl mnop…

라면 그것은 다음과 같이 된다.

…defghi hijklm lmnopq…

매 그룹의 바깥 두 비트들은 4개의 가능한 대입들중의 하나를 선택한다(S통의 한 행). 다음 4-bit출력값은 특정의 4-bit입력으로 대입된다(중간의 4개의 입력비트들). 8개의 S-통들로부터 32-bit출력이 치환되며 따라서 다음단계에서 매 S-통으로부터의 출력은 될수록 많은 다른것들에 영향을 준다.

열쇠생성

그림 3-7과 3-8을 보면 그 알고리듬의 입력에서 쓰이는 56bit열쇠는 PC-1이라는 표에 의하여 치환된다는것을 알수 있다(표 3-4의 ㄱ). 이 56bit는 C_0과 D_0이라는 표식이 붙은 두개의 28bit량들로 갈라서 처리된다. 매단에서 C_{i-1}과 D_{i-1}은 표 3-4의 C에서 주어진대로 왼쪽순환밀기되거나 1 혹은 2bit 회전된다. 이렇게 밀기된 값들은 다음 단의 입력으로 된다. 이 값들은 PC-2에 대한 입력으로 되는데 그것은 함수 $f(R_{i-1}, K_i)$에 대한 입력으로 공급되는 48bit출력을 생성한다.

표 3-4. DES열쇠계산에 리용되는 표

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

ㄱ) 치환된 선택 1(PC-1)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

ㄴ) 치환된 선택 2(PC-2)

단의 수	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
비트회전	1	1	2	2	2	2	2	2	1	22	2	2	2	2	1	

ㄷ) 왼쪽밀기의 표

DES의 복호화

페이스텔암호와 마찬가지로 복호화는 암호화와 같은 알고리듬을 리용하는데 부분열쇠의 적용순서는 반대이다.

사태효과

임의의 암호화알고리듬에서 얻으려는 성질은 평문이나 열쇠에서의 작은 변화가 암호문에서 큰 변화를 일으키는것이다. 특히 평문 혹은 열쇠에서 한 비트의 변화가 암호문의 많은 비트들에 변화를 주는것이 바람직하다. 만일 변화가 작다면 이것은 탐색해야 할 열쇠공간이나 평문공간의 크기를 줄이는 어떤 방법을 제공할수 있다.

DES는 강한 사태효과를 보여 준다. 표 3-5에 그 결과[KONH81]를 보여 주었다. 이 표에서 두 평문은 한 비트만 차이나는데 다음과 같다.

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

100000 00 00000000 00000000 00000000 00000000 00000000 00000000 00000000

열쇠는

00000001 1001011 0100100 1100010 0011100 0011000 0011100 0110010

표 3-5. DES에서 사태효과

(ㄱ) 평문에서 변화		(ㄴ) 열쇠에서 변화	
단	차이나는 비트수	단	차이나는 비트수
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

3개의 단들을 지난후 두 블로크는 21개의 비트들이 차이난다는것을 알수 있다. DES의 출력에서 두 암호문은 34개의 비트위치들에서 차이난다.

표 3-5의 ㄴ은 다음의 평문을 입력으로 하였을 때 검사결과를 주고 있다.

01101000 10000101 0010111 01111010 00010011 01110110 11101011 10100100

이때 두 열쇠는 한 비트의 위치에서만 다르다. 즉

1110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100

0110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100

암호문에서 대략 절반의 비트들이 차이나며 불과 몇단후에는 사태효과가 현저해 진다는것을 보여 준다.

3.4 DES의 강도

규격으로 제정된 후에도 DES에 의해 제공되는 보안수준과 관련한 문제들에 대하여 오래동안 론의가 진행되여 왔다. 이 론의는 크게 두가지 즉 열쇠의 크기와 알고리듬의 특성에 귀착된다.

56bit 열쇠의 사용

56bit길이의 열쇠인 경우 2^{56}개 즉 약 7.2×10^{16}개의 열쇠가 있을수 있다. 이것은 아마도 힘내기공격으로는 풀수 없을것이다. 평균적으로 열쇠공간의 절반을 탐색하면 된다고 보아도 1㎲당 한번의 DES암호처리를 진행하는 기계를 가지고 암호를 격파하는데 천년이상 걸릴것이다.

그러나 1㎲당 한번의 암호처리를 진행한다는 가정은 너무 소극적이다.

1977년에 디피(Diffie)와 헬만(Hellman)은 매 ㎲당 하나의 암호를 계산하는 100만개의 암호장치를 병렬기계로 구성할수 있다는 주장을 내놓았다. 매 암호장치는 1ms당 하나의 암호화를 수행한다. 이렇게 되면 평균탐색시간이 약 10시간 줄어 든다. 저자의 타산에 의하면 투자량은 1977년화페로 약 2000만$로 된다.

이 문제에 대하여 위너(Wiener)는 최근에 기지평문공격에 기초한 비교적 정확한 해석을 진행하였다. 이때 공격자는 적어도 하나의 평문과 암호문쌍을 가지고 있다고 가정하였다. 위너는 그 설계의 세부를 제공하는데 중점을 두었다. 위너에 의하면

> DES의 열쇠공간을 빨리 탐색하는 방법에 대한 확인할수 없는 수많은 주장들이 있다. 이러한 애매한 주장들을 피하기 위해 열쇠탐색기계의 설계에서 상당한 량의 세부들이 부록들에 포함되였다. 결과 DES의 열쇠를 찾기 위한 시간과 기계의 비용에 대한 정확한 타산을 얻게 되였다. 그러나 그러한 기계를 정확히 만들기 위한 계획은 세워 지지 않았다.

위너는 매초당 5000만개의 열쇠탐색속도를 달성하기 위한 관흐름기술을 리용하는 소편의 설계에 대하여 통보하였다. 1993년가격으로 비용은 10만$이고 5,760개의 열쇠탐색소편을 포함하는 모듈이 설계되였다. 이 설계에 의하여 다음의 결과들이 얻어 진다.

열쇠탐색기계단위비용	기대되는 탐색시간
10만$ 100만$ 1000만$	35h 3.5h 21min

그리고 1회개발비용은 약 50만$라고 평가하였다. 1997년 갱신안[WIEN]에서 같은 비용에 대하여 6개의 인자에 의하여 시간을 분할하였다(즉 10만$짜리기계의 기대되는 탐색시간은 6시간이다).

비록 위너의 연구는 의의는 있으나 그것은 가상적인 설계이며 아직 제작되지 않았다. DES의 더 심중한 약점은 비밀열쇠에 대한 RAS연구소의 도전을 통해서도 알수 있다. 그 도전은 1000$의 상금이 걸린것으로서 24개의 문자로 된 구 "the Unknown message is:"를 포함하는 세개의 알려 진 본문블로크와 그 뒤의 알려 지지 않은 평문통보문으로 이루어 진 평문에 대하여 암호문으로 주어 진 DES열쇠를 찾는것이였다. RAS는 1997년 1월 29일에 그 도전을 제기하였다. 그 도전에 응하여 로크버써(Rocke verser)는 힘내기공격프로그람을 개발하고 그것을 인터네트상에 공개하였다. 프로젝트는 인터네트상의 수많은 콤퓨터들과 련결되여 나중에는 70,000개이상의 체계를 보유하게 되였다. 매 새로운 참여자(DES분석에 참여하는 자)를 받으면 개발팀은 새 콤퓨터가 조사해야 할 DES열쇠공간의 새 부분을 할당하여 주었다. 체계는 1997년 2월 18일에 작업을 개시하여 모든 가능한 열쇠의 약 1/4을 조사한 다음 96일만에 정확한 열쇠를 찾았다. 이 도전은 견고한 암호학적문제를 공격하는데서 분산된 PC의 위력을 시위하였다.

그러나 여기에는 모든 가능한 열쇠를 단순히 한번 실행해 보는것외에도 열쇠탐색공격이 더 있다. 알려 진 평문이 제공되지 않는 한 분석자는 평문을 평문으로 인식할수밖에 없다. 만일 통보문이 영어로 된 본문이라면 결과는 쉽게 뽑아 내며 영어인식과제는 자동화된다. 또 본문통보문이 암호화전에 압축된다면 문제는 더 어렵게 된다. 또한 통보문이 수값화상과 같은 보다 일반적형태의 자료이고 압축되였다면 그 문제는 자동적으로 조종하기가 더 힘들어 진다. 힘내기공격방법을 보충하기 위해서는 예상되는 평문에 대한 어느 정도의 지식이 필요되며 자동적으로 평문을 선별해 내는 수단도 필요된다.

위너의 설계와 비밀-열쇠도전은 DES의 보안에 대한 수십년간의 론의의 절정을 이루며 어떤 의미에서는 전환점으로 되였다. 지금까지도 개인적 및 상업용응용에서는 DES를 리용할수 있다고 보고 있다. 그러나 전통암호를 새롭게 교체하는 문제를 조사해야 할 시대가 도래하였다. DES의 후보자로서 4장에서 론의되는 3중DES를 비롯하여 여러가지 새로운 전통암호알고리듬들이 제기되고 있다.

DES 알고리듬의 성질

또 하나의 론의점은 암호분석에서 DES알고리듬의 특징을 밝힐수 있다는 가능성이다. 론의의 초점은 매 반복에서 리용되는 S-통 혹은 8개의 대입표이다. 이 통들과 전체 알고리듬에 대한 설계기준이 공개되지 않았으므로 S-통에 대한 약점을 아는 적에 의한 암호분석이 가능하다고 볼수 있다. 이 가정을 시발점으로 하여 진행된 수십년동안의 연구조사과정에 수많은 규칙성과 예견치 못했던 거동들이 S통에서 발견되였다. 그럼에도 불구하고 S-통에서 가정되였던 치명적약점은 아직까지 해명되지 못하고 있다.

3.5 차분암호분석과 선형암호분석

DES는 그의 생명주기의 거의 전기간 열쇠길이가 비교적 짧기때문에 힘내기공격에 약하다는것이 기본문제이다. 그러나 DES에 대한 암호분석공격방법을 찾는것도 흥미 있다. 3중DES를 포함하여 긴 열쇠를 가지는 블로크암호의 보급이 증가됨에 따라 힘내기공격은 실천불가능하게 되였다. 그리하여 DES나 다른 대칭블로크암호에 대한 암호분석적공격들에 대한 문제가 더욱 중시되고 있다.

이 절에서는 두개의 가장 강력하고 기대되는 수법인 차분해석법과 선형해석법에 대한 간단한 개괄을 준다.

차분암호분석법

최근년간 암호분석에서 가장 눈에 띄우는 전진의 하나는 차분분석법이다. 이 절에서는 그 기술과 DES에 대한 적응성을 취급한다.

력사

차분암호분석법은 1990년까지 공개문헌에 소개되지 않고 있었다. 처음으로 발표된 성과는 무피(Murphy)에 의한 FEAL이라는 블로크암호분석이였다. 그후 비함(Biham)과 사미르(shamir)의 많은 론문들에서 다양한 암호알고리듬들과 하쉬함수에 대한 공격을 보여 주었는데 그 결과는 [BIHA93]에 요약되여 있다.

이 방법에 대한 대부분의 공개결과들은 DES에 응용되고 있는것들이다. 차분암호분석법은 2^{55}이하의 복잡성으로 DES를 격파할수 있는 최초의 공격이였다. 제안된 방식에 의해 2^{47}개의 선택평문을 요구하는 2^{47}정도의 로력으로 DES를 성공적으로 분석할수 있다. 2^{47}은 2^{55}보다 현저히 작지만 2^{47}개의 선택평문을 찾는 문제는 이 공격을 리론적흥미거리로만 만들어 버렸다.

비록 차분암호분석법이 강력한 도구라 할지라도 DES를 공격하는데는 적합치 않다. 그것은 DES를 설계한 IBM의 어떤 성원에 의하여 이 공격법이 벌써 1974년 초에 알려졌기때문이다.

차분암호분석을 리용하는 공격에 대처할데 대한 요구는 치환 P와 S-통설계에서 큰 역할을 놀았다. 이러한 변화의 영향을 보기 위해 증거로서 문헌[BIHA93]에 있는 비교결과를 고찰하자. 8단 LUCIFER알고리듬의 차분암호분석은 다만 256개의 선택평문만을 요구하며 DES의 8단판에 대한 공격은 2^{14}개의 선택평문을 요구한다.

IDEA의 설계에 영향을 미친 차분암호분석은 DES가 나온후 가장 중요한 전통암호의 하나로 되였다. IDEA의 설계자들은 잠정암호화표준(Proposed Encryption Standard: PES)으로 알려 진 초기설계를 내놓을 당시까지도 차분암호분석을 알지 못하였다.

PES에서는 128-bit열쇠가 사용되였는데 여전히 2^{64}정도의 복잡성준위를 가진 차분암호분석에 약하였다. 설계자들은 PES에 수정하여 내놓은 IDEA가 차분암호분석에 견딘다고 주장하고 있다.

차분암호분석공격

차분암호분석공격은 복잡한데 문헌[BIHA93]에서 충분히 소개되고 있다. 여기서는 이 공격에 대한 대체적표상을 줄수 있도록 개괄한다.

먼저 DES에 대한 표기에서 다음의 약속을 한다. 초기의 평문블로크의 두 반토막을 m_0, m_1라고 하자. DES의 매단에서 오른편 입력은 왼편 출력으로 넘어 가며 오른편 출력은 그 단에서의 부분열쇠와 함께 왼편 입력의 함수로 된다. 따라서 매 단에서 다만 1개의 32bit블로크만 창조된다. 만일 매 새 블로크를 $m_i (2 \le i \le m)$로 표시하면 중간통보문의 반부분들은 다음과 같이 련관된다.

$$m_{i+1}=m_{i-1} \oplus f(m_i, K_i), \qquad i=1, 2, \cdots, 16$$

차분암호분석에서 XOR차분이 $\Delta m = m \oplus m'$로 주어 진 두 통보문을, 그 다음에는 m과 m'로부터 시작하여 중간통보문반토막들의 차분 즉 : $\Delta m_i = m_i \oplus m_i'$를 고찰한다.

그러면

$$\begin{aligned} \Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\ &= [m_{i-1} \oplus f(m_i, K_i)] \oplus [m'_{i+1} \oplus f(m'_i, K_i)] \\ &= \Delta m_{i-1} \oplus [f(m_i, K_i) \oplus f(m'_i, K_i)] \end{aligned}$$

만일 같은 부분열쇠를 사용한다면 f에 대한 같은 차분을 가지는 많은 입력쌍들이 같은 출력차분을 준다고 가정한다. 이것을 더 정확히 고찰하기 위해 입력의 배타적론리합이 X인 쌍들의 분수 P에 대하여 출력의 배타적론리합 (XOR)가 Y와 같으면 X는 Y를 확률 P로 발생시킨다고 말한다. 특정의 출력차분을 발생시킬 확률이 큰 X의 값들이 몇개 있다고 가정할수 있다. 따라서 높은 확률을 가지는 Δm_{i-1}와 Δm_i를 안다면 높은 확률로 Δm_{i+1}를 알수 있다. 또 그러한 차분들이 여러개 결정된다면 함수 f에서 리용되는 부분열쇠를 결정하기 쉽다.

차분암호분석법의 총적전략은 한 단에 대한 이러한 고찰들에 기초하였다. 절차는 주어 진 차분을 가지는 두 평문통보문 m과 m'로 시작하여 암호문에 대하여 확률차분을 주는 매개 단다음의 차분의 확률패턴을 통하여 추적한다. 실제로 2개의 32bit반토막에 대한 2개의 확률차분 $(\Delta m_{17} \| \Delta m_{16})$이 있다. 다음에 미지의 열쇠에 관한 실제적차분을 결정하기 위하여 m과 m'에 암호화를 실시하고 확률차분과 결과를 비교한다. 만일 대조가

$$E_k(m) \oplus E_k(m') = (\Delta m_{17} \| \Delta m_{16})$$

와 같다면 모든 중간단들에서 모든 확률패턴들이 정확하다고 본다. 그러한 가정밑에서 열쇠비트에 대한 어떤 추론을 할수 있다. 이 절차는 모든 열쇠비트들을 결정할 때까지 여러번 반복해야 한다.

그림 3-10에 DES의 세개의 단들에서 차분의 전파과정을 보여 주었다. 오른쪽에 보여 준 확률들은 중간차분들의 주어 진 모임이 입력차분들의 함수처럼 나타날 확률이다. 총체적으로 세개의 단들을 거친후 출력차분의 확률은 보여 준바와 같이 $0.25 \times 1 \times 0.25 = 0.0625$이다.

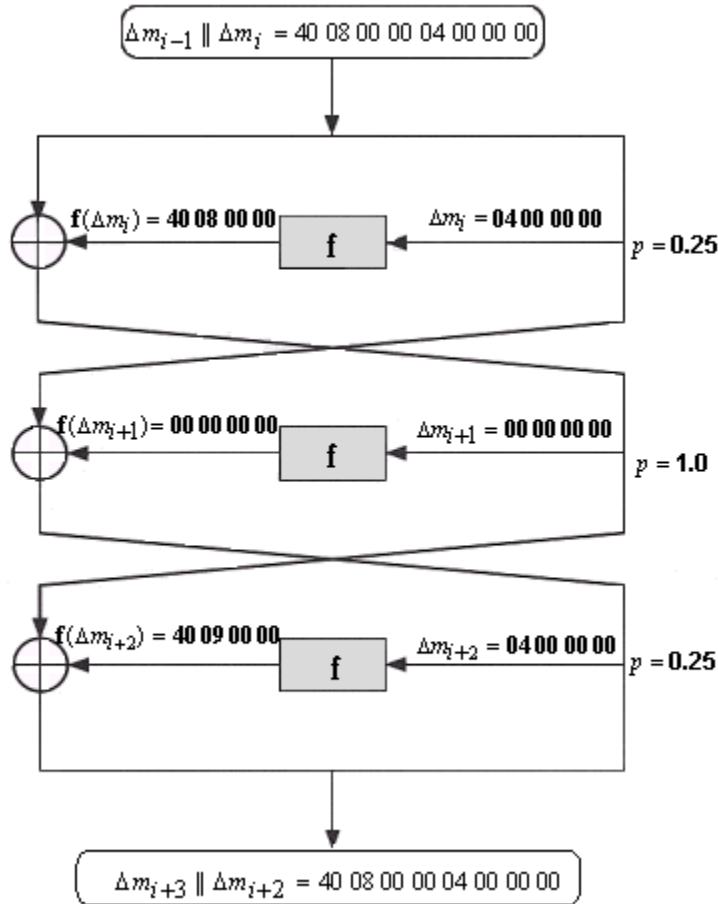



그림 3-10. DES의 3개의 단들을 통한 차분전파(16진수로)

선형암호분석

보다 최근에 개발된것이 문헌 [MATS93]에 소개된 선형암호분석이다. 이 공격은 DES에서 진행되는 변환들을 묘사하기 위한 선형근사를 찾는데 기초하고 있다. 이 방법에서는 차분암호분석에서의 2^{47}개의 선택평문에 비해 2^{47}개의 기지평문으로 주어 진 DES의 열쇠를 찾을수 있다. 이것은 기지평문을 얻는것이 선택평문을 얻는것보다 더 쉬울수 있으므로 차분암호분석보다 좀 발전했다고는 할수 있지만 이것으로 해서 선형암호분석은 여전히 DES에 대한 공격과 마찬가지로 공격 당할 위험이 있다.아직까지는 선형암호분석방법을 유효하게 하기 위한 연구사업이 그리 심화되지 못하고 있다.

아래에 선형암호분석법의 기초원리를 기본적으로 요약하였다. N-bit평문과 암호문블로크, m-bit열쇠를 가진 암호화에 다음과 같은 표기를 약속하자. 즉 평문블로크는 P[1], …, P[n]로, 암호문블로크는 C[1], …, C[n]로, 열쇠는 K[1], …, K[m]로 표기하자. 그다음

$$\mathrm{A}[i, j, \cdots k] = \mathrm{A}[i] \oplus \mathrm{A}[j] \oplus \cdots \oplus \mathrm{A}[k]$$

라고 정의하자.

선형암호분석은 확률 $p \neq 0.5$로 성립하는 효과적인 선형방정식 즉

$$\mathrm{P}[\alpha_1, \alpha_2, \cdots \alpha_a] \oplus \mathrm{C}[\beta_1, \beta_2, \cdots, \beta_b] = \mathrm{K}[\gamma_1, \gamma_2, \cdots, \gamma_c]$$

을 찾는데 귀착된다(여기서 x는 0 혹은 1; $1 \le a, b \le n$, $1 \le c \le m \le$ α, β, γ는 고정된 유일한 비트위치를 표시한다).

P가 0.5보다 클수록 더 효과적인 방정식으로 된다. 일단 어떤 관계가 결정되면 절차는 많은 평문암호문쌍들에 대하여 우의 방정식의 왼쪽의 결과들을 계산한다. 만일 결과들에 0이 절반이상 있다면 $\mathrm{K}[\gamma_1, \gamma_2, \cdots, \gamma_c] = 0$으로, 1이 절반이상이라면 $\mathrm{K}[\gamma_1, \gamma_2, \cdots, \gamma_c] = 1$로 한다. 이것은 열쇠비트들에 대한 선형방정식으로 된다. 이러한 관계식들을 더 많이 얻어 열쇠비트들을 구할수 있다. 선형방정식을 취급하고 있기때문에 문제는 결합된 결과를 가지고 한번에 암호의 한개의 단을 접근시킬수 있다.

3.6 블로크암호의 설계원리

암호학적으로 강한 블로크암호설계에서 큰 전진이 있었지만 기본원리는 1970년대 초에 DES와 페이스텔의 설계팀의 연구결과에 비해 그리 달라 진것이 없다. 그러므로 DES에서 리용되는것으로 소개되고 있는 설계규칙의 고찰로부터 시작한다. 블로크암호설계를 세개의 측면 즉 단의 수, F함수의 설계, 열쇠관리에 대하여 론의한다.

DES설계규칙

DES설계에 사용된 규칙은 문헌 [COPP94]에서 볼수 있는것처럼 S-통의 설계와 그의 출력을 취하는 P함수에 중심을 두었다(그림 3-9). S-통의 규격은 다음과 같다.

1. 임의의 S-통의 출력비트는 입력비트들의 선형함수에 너무 근사해서는 안된다. 특히 임의의 출력비트와 임의의 6개 입력비트들의 부분모임을 선택하는 경우 출력비트가 그 입력비트들의 XOR와 같을 입력들의 비는 0 혹은 1이 아니라 1/2에 가까와야 한다.
2. S-통의 매행은(맨 왼쪽과 맨 오른쪽 비트값에 의하여 결정되는) 모두 16개의 가능한 출력비트조합들을 포함하여야 한다.
3. 만일 S-통에 대한 두 입력이 정확히 한 비트에서 차이나면 출력은 적어도 두 비트에서 차이나야 한다.
4. 만일 S-통에 대한 두 입력이 가운데 있는 두 비트들에서 차이난다면 출력은 적어도 두 비트에서 차이나야 한다.
5. 만일 S-통에 대한 두 입력들이 첫 두 비트들에서 차이나고 그의 두 마지막비트들이 일치한다면 그 두 출력들은 같지 말아야 한다.
6. 입력들사이의 임의의 령아닌 6-bit차분에 대하여 그 차분을 표시하는 32개의 입

력쌍중에서 8개 이상은 같은 출력차분을 만들지 말아야 한다.

7. 이 규칙은 3개의 S-통의 경우를 내놓고는 앞의 규칙과 비슷하다.

코퍼스미스(Coppersmith)는 우의 규칙중에서 첫째 항목만이 DES에서 S-통들만이 비선형부분이므로 필요하다고 지적하였다. 만일 S-통들이 선형이라면(즉 매 출력비트는 입력비트의 선형결합으로 된다.) 전체 알고리듬이 선형이 되고 쉽게 격파되게 된다. Hill암호가 선형이므로 쉽게 격파된다는것을 이미 보았다. 나머지 항목에 대해서는 차분암호분석을 막고 충분한 혼란성을 가지게 하는데 선차적목적을 두었다.

치환 p를 위한 기준은 다음과 같다.

1. 단 i에서 매 S-통의 4개의 출력비트들은 그것들중 두개가 $(i+1)$단의 가운데비트들에 영향을 주며(가운데비트들에 대한 입력) 다른 두 비트들은 끝비트들에 영향을 미치도록 분산된다. S-통에 대한 입력의 두 가운데비트들은 린접한 S-통과 공유되지 않는다. 끝비트들은 왼쪽의 두 비트와 오른쪽의 두 비트들인데 그것은 린접한 S-통과 공유된다.
2. 매 S-통에서 4개의 출력비트들은 다음단계의 6개의 서로 다른 S-통들에 영향을 주며 어느 두 비트도 같은 S-통에 영향을 주지 않는다.
3. 두 S-통 j, k에 대하여 S_i의 어느 출력비트도 다음단에서 S_k의 가운데비트에 영향을 미치지 않으면 S_k의 출력비트는 S_i의 가운데비트에 영향을 미칠수 없다. 이것은 j=k에 대하여 S_i의 출력비트가 S_i의 가운데비트에 영향을 줄수 없다는것을 의미한다.

   이 항목들은 알고리듬의 혼란성을 증대시키는데 기본목적을 두고 있다.

단의 수

페이스텔암호의 강도는 설계의 세 측면 즉 단의 수, 함수 F, 열쇠관리알고리듬에 의해 결정되는데 그중에서 단의 수에 대하여 먼저 보자.

단의 수가 크면 상대적으로 F가 약해도 암호분석이 어렵게 된다. 일반적으로 이 기준은 알려 진 암호분석이 단순한 힘내기공격보다 더 큰 품이 요구되게끔 단의 수를 선택할것을 요구한다. 이 기준은 DES의 설계에 일정하게 리용되였다. 스네어(schneier)[SCHN96]은 16단 DES를 고찰하고 차분암호분석공격은 힘내기공격보다 효과적이 못된다는것을 지적하였다. 차분암호분석공격은 $2^{55.1}$의 복잡성을 요구하지만 전수공격은 2^{55}의 복잡성을 요구한다. 만일 DES가 15이하의 작은 단수를 가진다면 차분암호분석은 힘내기공격보다 더 효률적일것이다.

이 기준은 어떤 알고리듬의 강도를 쉽게 판정하고 서로 다른 알고리듬들의 비교를 쉽게 하므로 의의가 보다 크다. 암호분석에서 좋은 결과가 없을 때 우의 기준을 만족하는 임의의 알고리듬의 강도는 열쇠길이만을 가지고 판정할수 있다.

함수 F의 설계

페이스텔블로크암호의 《심장》은 함수 F이다. DES에서 이 함수는 S-통의 리용에 의거한다. 이것은 4장에서 고찰하게 되는 모든 다른 대칭암호의 경우에도 마찬가지이다. 그러나 함수 F를 설계하기 위한 기준에서 몇가지 일반적인 론의가 필요하다. 그 다음 S-통설계를 특별히 취급한다.

함수 F의 설계기준

함수 F는 페이스텔암호에서 혼란요소를 제공한다. 따라서 F에 의하여 수행되는 대입을 해석하기가 어려워야 한다. 하나의 명백한 기준은 F가 비선형이여야 한다는것이다. F가 보다 비선형일수록 임의의 암호분석은 더 어려워 진다. 비선형성을 재는 여러가지 측도가 있지만 간단히 말하면 F를 어떤 선형방정식들로 근사시키기 어려울수록 F의 비선형성은 크다.

F의 설계에서는 여러가지 다른 기준들을 고찰할수 있다. 알고리듬은 사태성질이 좋아야 한다. 다시말하여 이것은 입력에서 한 비트의 변화가 출력에서 여러 비트의 변화를 일으킨다는것을 의미한다. 이에 대한 더 엄격한 정의는 엄격한 사태판정(SAC)[WEBS86]인데 그것은 임의의 하나의 입력비트 i가 모든 i, j에 대해 전도될 때 어떤 S-통의 임의의 출력비트 j가 확률 1/2로 변화된다는것을 말한다. SAC가 S-통들로 표현되지만 이와 같은 기준을 총괄적으로 F에 적용할수 있다. 이것은 S-통을 포함하지 않는 설계를 고찰할 때 중요하다.

[WEBS86]에 제안된 다른 기준으로 비트독립기준(blt indepedence criterion-BIC)을 들수 있는데 그것은 하나의 입력비트 *i*가 전도될 때 출력비트 j와 K가 임의의 모든 i, j에 대하여 독립적으로 변환될것을 요구한다. SAC와 BIC기준은 혼란함수의 효과성을 높이는것으로 된다.

S-통설계

대칭블로크암호분야에서 연구가 가장 심화되고 있는 문제가 S-통설계이다. 그에 대한 론문들이 수없이 발표되고 있다. 여기서는 그중 몇가지 일반적원리들만을 보기로 한다. 요컨대 S-통에 대한 입력벡토르의 임의의 변화가 출력에서는 어떤 무질서한 변화를 일으키도록 하려고 한다고 하자. 그 관계는 비선형으로 되여야 하며 선형함수로 조사하기 어려워야 한다.

S-통의 하나의 명백한 지표는 그 통의 크기이다. n×m S-통은 n개의 입력비트들과 m개의 출력비트를 가진다. DES는 6×4 S-통을 가진다. 4장에서 취급하는 Blowfish와 CAST는 8×32 S-통을 가진다. S-통이 클수록 차분 및 선형해석에 더 잘 견딘다 [SCHN96]. 다른 한편 n이 클수록 큰 표를 취급하여야 한다. 실천적측면에서 n의 크기는 보통 8~10사이로 제한된다. 또한 S-통이 클수록 실천적에서는 그 통을 정확히 설계하기가 더 어려워 진다.

S-통들은 일반적으로 DES에서 리용된 방법과는 달리 조직화된다. n×m S-통은 일반적으로 m bit들이 들어 있는 2^n개의 행으로 이루어 진다. 입력의 n개 비트들은 그 S-통의 행중의 하나를 선택하며 그 행에서 m bit들이 출력으로 된다. 실례로 8×32 S-통에서 입력이 00001001이면 출력은 9행(첫행은 0행이다.)의 8bit로 이루어 진다.

미스터(Mister)와 아담스(Adams)는 S-통설계와 관련한 여러개의 기준을 제안하였다. 그중에는 S-통이 SAC와 BIC를 다 만족시켜야 한다는것도 들어 있다. 또 그들은 S-통의 렬들의 모든 선형결합이 bent이여야 한다고 보았다. 일정한 수학적기준에 따르면 bent함수들은 비선형성이 강한 론리함수들(Boolean)의 특수한 부류이다. 이에 대한 간단한 소개를 부록 3A에 주었다. Bent함수의 리용에 기초한 S-통의 설계와 해석에 대한 연구가 심화되고 있다.

S-통과 관련된 기준들은 문헌 [MEYS95]에서 분석되고 있다. 담보된 사태(Guaranteed Avalanche: **GA**)기준은 다음과 같다. S-통은 다음과 같은 경우에 차수 γ의 GA를 만족한다. 입력에서 1-bit의 변환에 대하여 출력에서 적어도 γ개의 출력비트들에서 변화를 일으킨다. 연구결과(저자의)는 γ가 2-5범위일 때 전반적인 암호알고리듬에서 강한 혼란성이 나타난다는것을 보여 주고 있다.

큰 S-통(례하면 8×32와 같은)인 경우 그 입력을 선택하는 문제가 여러가지로 론의

되고 있다. 니베그(Nyberg)는 다음의 방법들을 제기하였다.

- **불규칙적인 방법:** S-통에 대한 입력을 주기 위하여 모조란수발생기 혹은 란수ㄱ표를 리용한다. 작은 S-통(례하면 6×4 등)에 대해서는 좋지 못한 결과를 줄수도 있으나 큰 S-통(례하면 8×32)에 대해서는 좋은 결과를 준다.
- **시험을 동반하는 불규칙적인 방법:** S-통의 입력을 우연선택하고 각이한 기준에 대한 그 결과를 검사하고 통과할수 없는것은 버린다.
- **사람이 만드는 방법(Man-Made):** 이것은 그것을 지원하는데 단순한 수학적지식만을 리용하는 다소 수동적인 DES설계방법인데 큰 S-통에서는 어렵다.
- **수학적으로 만드는 방법:** 이것은 수학적원리에 따라 S-통을 발생시키는 방법이다. 수학적원리를 리용하여 충분한 혼란성을 가지면서 선형 및 차분암호분석에 견디도록 구축할수 있다. 이 수법은 4장에서 취급한 CAST와 함께 리용되는 방법이다.

첫 방법은 우연성과 열쇠의존성을 다 가지는 S-통을 리용하는것이다. 이 방법의 실례가 4장에서 취급되는 Blowfish인데 거기서는 모조란수자들로 채운 S-통으로부터 시작하여 열쇠를 리용하여 내용을 변경한다. 열쇠의존 S-통의 중요한 우점은 통의 내용이 고정되지 않으므로 약점을 알기전에는 S-통을 해석할수 없다는것이다.

열쇠관리알고리듬

블로크암호설계의 최종령역이며 S-통의 설계보다 부차적인것으로 되고 있는것이 열쇠관리알고리듬이다. 페이스텔블로크암호에서 열쇠는 매 단에서 하나의 부분열쇠를 발생시키는데 쓰인다. 일반적으로 개별적인 부분열쇠를 추론하기가 최대한으로 어렵게끔 부분열쇠를 선택하며 더우기 기본열쇠에 대한 추측은 더욱 어렵게끔 하여야 한다. 발표된 일반적원리는 아직 없다.

할(Hall)은 최소한 열쇠관리는 열쇠 및 암호문의 엄격한 사태기준(SAC)과 비트독립기준 BIC를 담보하여야 한다는것을 지적하였다[ADAM94].

3.7 블로크암호의 동작방식

DES알고리듬은 자료보안을 위한 기초블로크이다. 각이한 응용에 DES를 적용하기 위하여 4개의 동작방식이 정의되였다. 이 4개의 동작방식들에 DES를 리용할수 있는 모든 가능한 암호응용들이 포함되도록 하였다. 이 방식들은 아래의 표에 요약되여 있으며 이 절의 나머지부분에서 간단히 취급되였다. 이와 같은 방식들은 임의의 대칭블로크암호에서도 적용될수 있다.

전자부호책방식

가장 단순한 방식은 전자부호책방식(Electronic Codebook Mode: ECB)인데 평문은 한번에 64bit씩 취급되며 그의 매 블로크는 같은 열쇠를 리용하여 암호화된다(그림 3-11). 암호책이라는 용어를 사용한 리유는 주어 진 열쇠에 대하여 평문의 매 64bit블로크당 유일한 암호문이 있기때문이다. 그러므로 모든 64bit평문패턴에 대응한 암호문을 보여 주는 입구를 지적하는 거대한 암호책을 생각할수 있다.

64bit보다 더 긴 통보문인 경우 통보문을 64bit블로크로 갈라 놓는다. 이때 마지막

블로크는 메꾸기(padding)를 할수 있다. 복호는 한번에 한 블로크씩 같은 열쇠에 의하여 수행된다. 그림 3-11에서 평문은(필요에 따라 메꾸기되는) 64bit의 블로크 $P_1, P_2, \cdots, P_N$로 구성되며 대응하는 암호문은 $C_1, C_2, \cdots, C_N$이다.

표 3-6. DES의 동작방식

방식	서술	일반적인 응용
전자부호책(Electronic Codebook:ECB)	매 64bit평문블로크는 같은 열쇠를 리용하여 독립적으로 암호화된다.	• 단순한 값들의 안전한 전송(실례로 암호열쇠)
암호블로크련쇄 (Cipher Block Chaining: CBC)	암호알고리듬에 대한 입력은 평문의 다음 64bit의 평문과 암호문의 선행한 64bit의 XOR이다.	• 범용블로크-지향전송 • 인증
암호문반결합 (Cipher FeedBack: CFB)	입력은 한번에 J개 비트씩 처리된다. 선행한 암호문은 암호알고리듬의 입력으로 들어 가 준우연출력을 생성하는데 그것은 평문과 XOR로 되여 다음암호문을 만든다.	• 일반목적흐름지향전송 • 인증
출력반결합 (Output Feedback: OFB)	암호알고리듬에 대한 입력이 선행한 DES출력이라는것을 제외하면 CFB와 비슷하다.	• 잡음통로우에서 흐름지향전송(례로 위성통신)

ㄱ) 암호화

ㄴ) 복호화

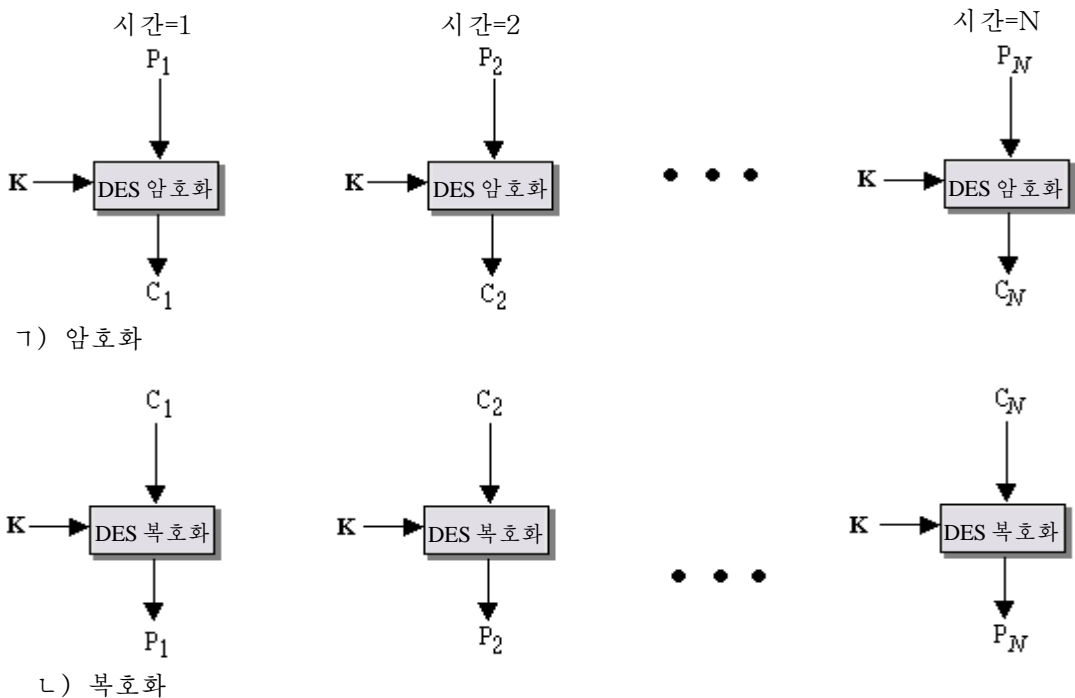



그림 3-11. 전자부호책(ECB)방식

ECB방법은 암호열쇠와 같이 자료량이 적을 때 리상적이다. 따라서 DES열쇠를 비밀로 전송하는데서 ECB가 적당한 방식으로 된다.

ECB의 가장 중요한 특징은 같은 64bit의 평문블로크가 통보문중에 한번이상 나타나는 경우 늘 같은 암호문을 생성한다는것이다.

긴 통보문에 대하여 ECB방식은 안전하지 못하다. 만일 통보문이 고도로 구조화되였다면 암호분석자들이 그 규칙성을 찾아 낼수 있다. 실례로 통보문이 늘 미리 정의된 어떤 마당으로 시작된다면 암호분석자는 많은 기지의 평문-암호문쌍을 가지고 작업할수 있다. 만일 그 통보문이 반복적인 요소를 가지고 있다면 그 요소는 해석자에 의하여 동정될수 있다. 이것은 해석을 도와 주거나 블로크를 대입 혹은 재배렬할수 있는 기회를 줄수 있다.

암호블로크련쇄방식

ECB의 보안상부족점을 극복하는데서 같은 평문블로크가 반복되는 경우 다른 암호블로크를 생성하는 기술을 도입하는것이 좋다. 이 요구를 만족시키기 위한 간단한 방법이 암호블로크련쇄방식이다(그림 3-12).

이 방식에서 암호알고리듬에 대한 입력은 현재의 평문블로크와 선행한 암호문블로크의 XOR이다. 매 블로크에서 같은 열쇠가 사용된다. 요컨대 평문블로크렬의 처리는 모두 련쇄를 이룬다. 매 평문블로크에 대한 암호화함수의 입력은 그 평문블로크와 고정된 관계를 가지지 않는다. 따라서 64bit의 반복되는 패턴들은 로출되지 않는다.

복호를 위하여 매 암호블로크는 복호알고리듬을 거친다. 그 결과는 선행한 암호문블로크와 XOR되여 평문블로크를 생성한다. 이 과정을 보기 위하여 다음과 같이 쓸수 있다.

$$C_n=E_k[C_{n-1}\oplus P_n]$$

그러면

$$D_K[C_n]=D_K[E_K(C_{n-1})\oplus P_n)]$$
$$D_K[C_n]=(C_{n-1}\oplus P_n)$$
$$C_{n-1}\oplus D_K[C_n]=C_{n-1}\oplus C_{n-1}\oplus Pn=P_n$$

암호문의 첫 블로크를 생성하기 위하여 초기화벡토르(IV)가 평문의 첫 블로크와 XOR된다. 복호화에서 IV는 복호알고리듬의 출력과 XOR되여 평문의 첫 블로크를 재생한다.

IV는 송신자와 수신자 량측에 다 알려 져야 한다. 최대의 보안을 위하여 IV는 열쇠와 마찬가지로 보호되여야 한다. 이것은 ECB암호화를 리용하여 IV를 보내면 된다. IV를 보안해야 할 하나의 근거는 다음과 같다. 만일 적이 IV의 다른 값을 리용하여 수신자를 속이려 한다면 평문의 첫 블로크에서 선택된 비트들을 바꾸어 놓을수 있다. 이에 대하여 다음과 같은 관계를 보자.

$$C_1=E_K(IV\oplus P_1)$$
$$P_1=IV\oplus D_K(C_1)$$

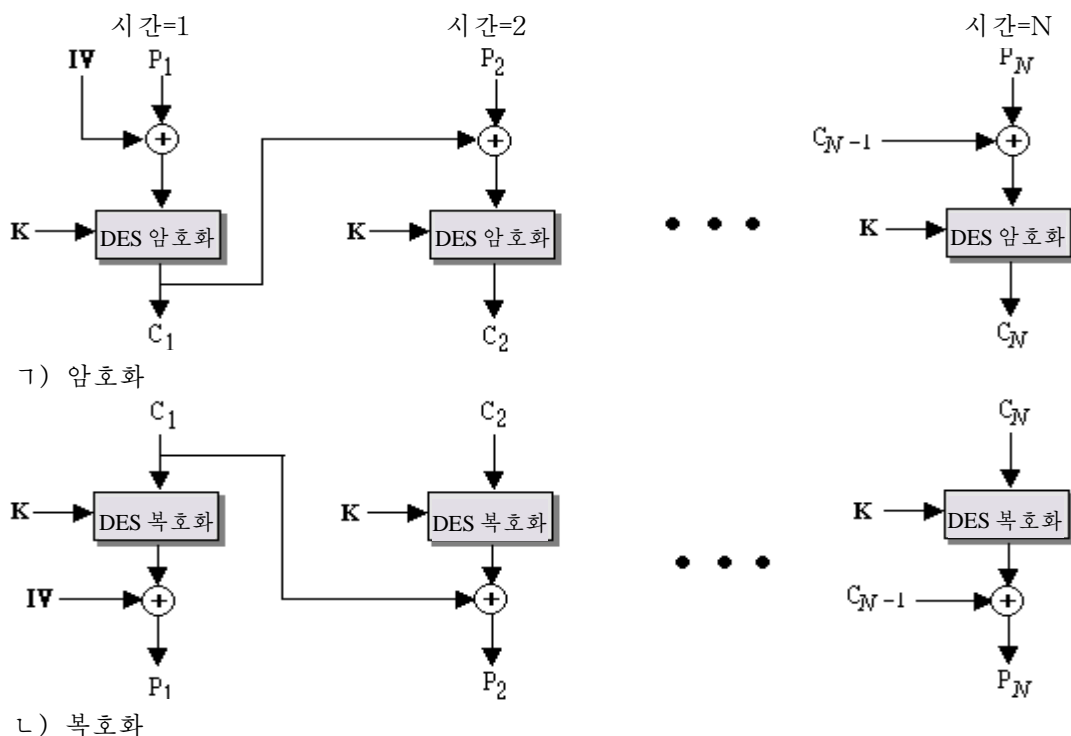



그림 3-12. 암호블로크련쇄방식

이제 $X[i]$로 64-bit량 X의 i번째 비트를 표기하자. 그러면

$$P_1[Pi]=IV[i] \oplus D_K(C_1)[i]$$

이로부터 XOR의 성질을 리용하여 다음과 같이 표시할수 있다.

$$P_1[i]=IV[i]' \oplus D_K(C_1)[i]$$

여기서 웃반점이 붙은 표기는 대응하는 비트가 반전되였음을 의미한다. 이것은 만일 적이 IV의 비트들을 예측적으로 변화시킬수 있다면 수신된 P_1의 값에 대응하는 비트들이 변경될수 있다는것을 의미한다.

IV의 지식에 기초한 다른 공격들에 대하서는 문헌 [VOYD83]을 참고할수 있다.

결론적으로 CBC기구의 련쇄는 64bit보다 더 큰 길이의 통보문을 암호화하는데 적당한 방식이다.

그리고 기밀성을 달성하는 외에도 CBC방식은 인증에 리용할수 있다. 8장에 그 리용에 대하여 서술되여 있다.

암호반결합방식

DES방식은 본질적으로 64bit블로크를 리용하는 블로크암호기술이다. 그러나 암호반

결합이나 출력반결합방식을 리용하여 DES를 흐름암호로 전환시킬수 있다. 흐름암호는 DES암호에서 블로크가 64bit로 되도록 메꾸기하던것을 하지 않아도 된다. 또한 실시간

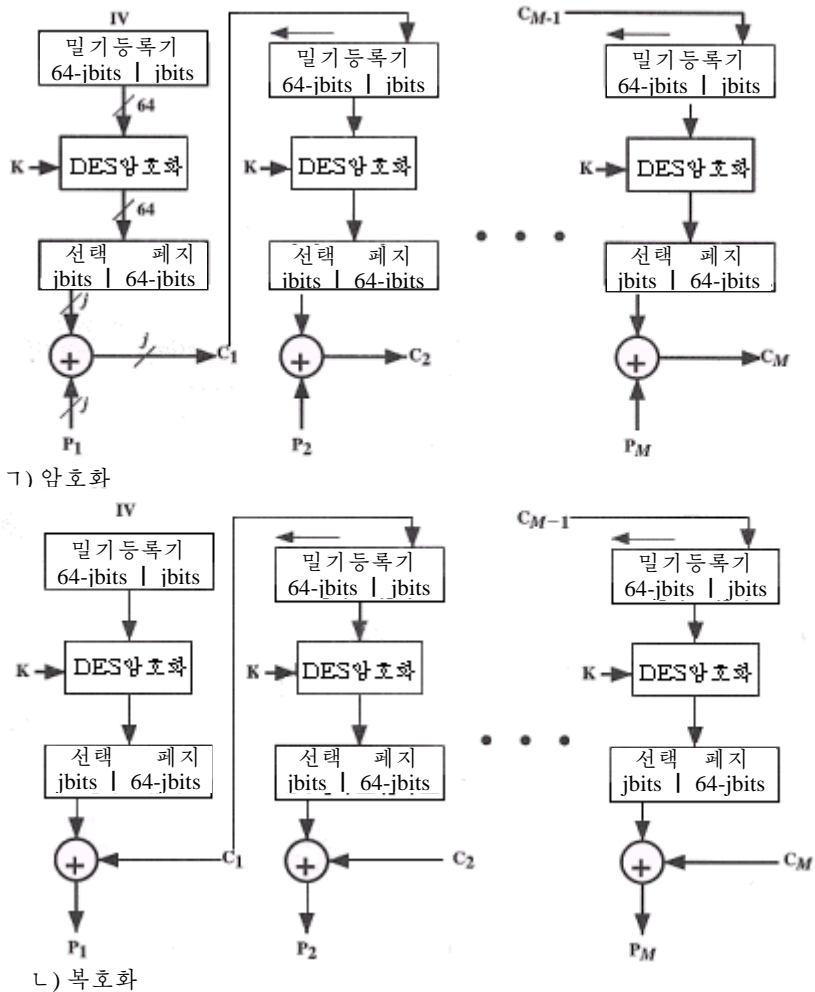



그림 3-13. Jbit암호반결합(CFB)방식

적으로 조작할수 있다. 그러므로 기호흐름을 전송할 때 매 기호는 암호화되며 기호-지향 흐름암호에 의하여 즉시에 전송될수 있다.

흐름암호가 가져야 할 성질의 하나는 평문과 같은 길이의 암호문을 가져야 한다는것이다. 즉 8bit짜리기호들이 전송되는 경우 매 기호들은 8개 비트들로 암호화되여야 한다. 만일 8개 이상의 비트들이 사용된다면 전송용량이 랑비된다.

그림 3-13에 CFB방식을 주었다. 그림에서 전송단위는 jbit라고 가정하였는데 일반적으로 j=8이다. CBC와 마찬가지로 평문의 단위들은 서로 련쇄를 이루며 따라서 임의의 평문단위에 대한 암호문은 모든 선행한 평문의 함수이다.

먼저 암호화를 고찰하자. 암호함수에 대한 입력은 처음에 어떤 초기화벡토르(IV)가 설정된 64-bit밀기등록기이다. 암호화함수의 출력의 맨 왼쪽의 j개 비트들은 평문의 첫

단위 P_1와 XOR되여 암호문 C_1를 만들고 다음에 C_1가 전송된다. 밀기등록기의 내용은 jbit만큼 왼쪽으로 밀기되며 C_1는 밀기등록기의 맨 오른쪽의 j개 비트들에 들어 있게 된다. 이 과정은 모든 평문의 단위들이 암호화될 때까지 계속된다.

복호화의 방식도 같은데 다만 접수된 암호문으로부터 평문을 생성하기 위하여 암호문이 암호화함수의 출력으로 XOR되는것이 다르다. 사용되는것은 복호화함수가 아니라 암호화함수임을 주의해야 한다. 이것은 쉽게 설명할수 있다. $S_j(X)$를 X의 가장 왼쪽의 j개의 비트라고 하자. 그러면

$$C_1=P_1\oplus S_j(E(IV))$$

따라서

$$P_1=C_1\oplus S_j(E(IV))$$

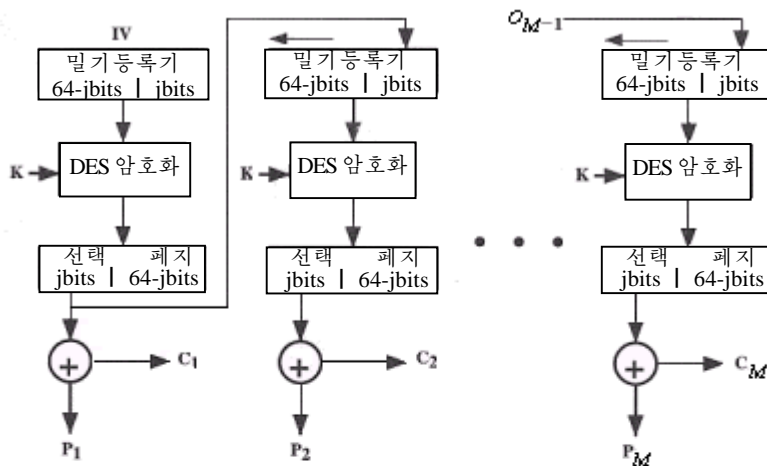
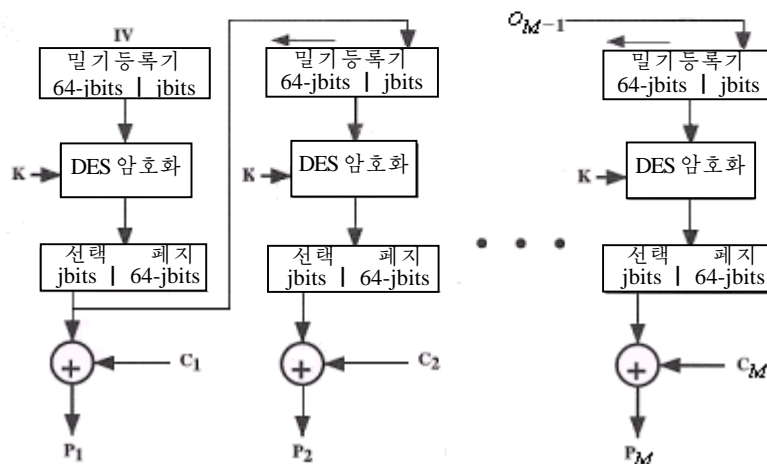



그림 3-14. J-bit출력반결합방식(CFB)

동일한 추론과정이 처리의 전과정에 반복된다.
또한 기밀성을 위해 CFB방식이 인증의 수단으로 리용될수 있다. 이 과정을 8장에서 취급한다.

출력반결합방식

출력반결합방식은 그림 3-14에 제시한것처럼 구조상 CFB와 류사하다. OFB에서 밀기등록기에 반결합하는것은 암호화함수의 출력이지만 CFB에서는 암호문단위가 밀기등록기에 반결합된다.

OFB방법의 우점은 전송시에 오유비트들이 전파되지 않는것이다. 실례로 C_1에서 어떤 오유비트가 생겨도 다만 복호된 P_1에만 영향을 주며 다른 평문단위들에는 영향을 주지 않는다. CFB에서 C_1는 밀기등록기의 입력으로 되며 따라서 오유가 계속 전파된다.

OFB의 결함은 CFB보다 통보문변경공격에 약하다는것이다. 암호문에서 한 비트를 반전시키는것은 회복한 평문에서 대응하는 비트를 반전시킨다. 따라서 회복된 평문에 대한 변경을 조종할수 있다. 이러한 변경은 적이 할수도 있는데 적들은 자료부분을 변경시키거나 필요하면 통보문의 검열합부분을 변경하거나 오유교정코드로는 걸리지 않는 방법으로 암호문을 변경시킬수 있다. 그에 대한 구체적인 내용은 [VOYD83]에서 참고하시오.

참고문헌

전통암호와 관련한 참고서가 많은데 그중에서 중요한것들을 소개하면 다음과 같다.

BARK91 Barker, W. *Introduction to the Analysis of the Data Encryption Standard (DES)*. Laguna Hills, CA: Aegean Park Press, 1991.

COPP94 Coppersmith, D. “The Data Encryption Standard (DES) and Its Strength Against Attacks.” *IBM Journal of Research and Development*, May 1994.

MEME97 Menezes, A.; Oorshcot, P.; and Vanstione, S. *Handbook of Applied Cryptography*.
Boca Raton, FL: CRC Press, 1997.

SCHN96 Schneier, B. *Applied Cryptography*. New York: Wiley, 1996.

SIMM92 Simmons, G., ed. *Contemporary Cryptology: The Science of Information Integrity*.
Piscataway, NJ: IEEE Press, 1992.

WIEN93 Wiener, M. "Efficient DES Key Search." *Proceedings, crypto’93*, 1993; Published by Springer-Verlag.

문 제

1. S-DES에 대한 열쇠생성을 보여 주는 그림 3-2를 참고하여 대답하시오.
   1) 초기 P10치환함수는 얼마나 중요한가?
   2) 두개의 LS-1밀기함수는 얼마나 중요한가?
2. S-DES에서 변수 q와 r에 대한 방정식들이 S-DES해석에 대한 절에서 정의되였다. S와 t에 대한 방정식을 만드시오.
3. S-DES를 리용하여 열쇠 (0111111101)을 사용하여 렬 (10100010)을 수동적으로 복호하시오. 매 함수 (IP, F_K, SW, F_K, IP^{-1})를 수행한후의 중간결과를 제시하시오. 그리고 평문문자렬의 첫 4bit들을 어떤 문자로 복호하고 두번째 4bit들을 다른 문자로 복호하시오. 여기서 2진수로 A부터 P까지의 문자들을 부호화하였다 (즉 A=0000, B=0001, …, P=1111). 암시: 중간검사로서 SW의 적용후에 렬은 (00010011)로 되여야 한다.
4. π를 옹근수 $1,2,\cdots,2^n-1$의 어떤 치환이라고 하자. 여기서 $\pi(m)$은 $m, 0 \le m \le 2^n$의 값의 치환을 준다.다시말하여 π가 n-bit 옹근수들의 모임을 자기자신에로 넘기고 어느 두 옹근수도 같은 옹근수에로 넘겨 지지 않는다. DES는 그와 같은 64bit옹근수의 치환이다. 만일 $\pi(m)=m$이면 π는 부동점을 가진다고 말한다. 즉 π가 어떤 암호화넘기기라면 부동점은 자지자체로 암호화되는 통보문이다. π가 부동점을 가지지 않을 확률이 중요하다. 넘기기의 60%이상이 적어도 하나의 부동점을 가질수 있는 예상밖의 결과를 보여 주시오.
5. 길이 n의 블로크를 암호화하는 블로크암호화알고리듬을 생각하자. 이때 $N=2^n$이라고 정의하자. t개의 평문-암호문쌍 P_i, C_i (여기서 $C_i=E_K[P_i]$)가 주어 졌다고 하자. 열쇠 K는 N!개의 가능한 넘기기중에서 선택된다고 가정하자. 힘내기공격(완전탐색)으로 열쇠 K를 탐색하려고 한다. 이를 위해 열쇠 K′를 발생시키고 $C_i=E_{K'}[P_i]$가 성립하는가를 $1 \le i \le t$에 대하여 검사한다. 만일 K′가 매 P_i를 그것의 정확한 C_i로 암호화하면 K′=K라고 본다. 그러나 $E_K(\bullet)$와 $E_{K'}(\bullet)$가 꼭 t개의 쌍에서는 일치하고 그밖의 모든 쌍들에서는 일치하지 않는 경우가 있을수 있다.
   ㄱ) $E_K(\bullet)$와 $E_K(\bullet)$가 실지로 서로 다른 넘기기일 확률은 얼마인가?
   ㄴ) $E_K(\bullet)$와 $E_K(\bullet)$가 다른 t'개 평문-암호문쌍 $(0 \le t' \le N-t)$에서 일치할 확률은 얼마인가?
6. DES의 복호화는 사실상 DES의 암호화의 거꿀과정임을 밝히시오.
7. DES알고리듬의 16번째 반복후에 32bit교환은 암호문을 거꿀순서로 된 열쇠를 가지고 알고리듬에 암호문을 반대로 단순히 실행시키여 암호화과정을 거꾸로 하는데 필요하다. 이것을 문제 3.6에 주었다. 그러나 왜 32bit교환이 필요한가는 아직 석연치 않다. 그것을 명백히 하기 위해 다음 문제를 푸시오. 먼저 몇가지 표기를 약속하자.

   A||B= 기호렬 A와 B의 련결
   T_i(R||L)= 암호화알고리듬의 i번째 반복에서 정의되는 변환, $1 \le i \le 16$
   TD_i(R||L)= 복호화알고리듬의 i번째 반복에서 정의되는 변환, $1 \le i \le 16$
   T_{17}(R||L)= L||R: 이 변환은 암호화알고리듬의 16번째 반복후에 일어 난다.

1) $TD_1(IP(IP^{-1}(T_{17}(T_{16}(L_{15}||R_{15})))))$는 32-bit반토막 L_{15}와 R_{15}의 교환변환과 동등함을 밝히시오. 즉

$$TD_1(IP(IP^{-1}(T_{17}(T_{16}(L_{15}||R_{15})))))= R_{15}||L_{15}$$

2) 암호화알고리듬에서 마지막 32-bit교환부분을 제거하였다고 가정하자. 이때 다음 관계가 성립하는가?

$$TD_1(IP(IP^{-1}(T_{16}(L_{15}||R_{15}))))= L_{15}||R_{15}$$

8. 표 3-2의 ㄱ와 표 3-4의 ㄴ를 비교하시오. 구조가 류사한가. 만일 류사하다면 그 류사성을 설명하시오 여기로부터 어떤 결론을 내릴수 있는가?
9. 복호를 위해 DES알고리듬을 리용한다면 16개의 열쇠($K_1, \cdots, K_{16}$)들이 거꿀순서로 적용된다. 따라서 그림 3-8의 오른쪽은 필요 없다. 복호처리를 위한 적당한 밀기절차를 가지는 (표 3-4의 ㄷ와 류사한)열쇠창조도식을 설계하시오.
10. ㄱ) M′는 M의 매 비트를 반전한것이라고 하자. 만일 평문블로크의 보수를 취하고 또 암호열쇠도 보수를 취한다면 이 값을 가지고 진행한 암호화의 결과는 초기암호문의 보수라는것을 증명하시오. 즉

만일

$$Y=DES_K(X)$$

이면

$$Y'=DES_{K'}(X')$$

암시: 길이가 같은 임의의 두 비트렬 A와 B를 가지고 $(A \oplus B)' = A' \oplus B$ 임을 보여주시오.

ㄴ) DES에서 힘내기공격에 필요한 열쇠공간의 크기는 2^{56}이다. ㄱ의 결과는 그것을 변화시키는가?

11. DES에서 매 부분열쇠의 첫 24bit는 초기열쇠의 28bit의 같은 부분모임에서 생기며 매 부분열쇠의 두번째 24bit들은 초기열쇠의 28bit의 다른 부분모임에서 생긴다는것을 밝히시오.
12. DES의 ECB방식으로 전송된 암호문의 어떤 블로크에 오유가 있으면 대응하는 평문블로크만이 영향을 받는다. 그러나 CBC방식에서는 이 오유가 전파된다. 실례로 전송된 C_1에 있는 오유는(그림 3-12) 명백히 P_1와 P_2에 영향을 미친다.

ㄱ) P_2외에 어떤 블로크에 영향을 미치는가?

ㄴ) 가령 P_1의 원천에서 어떤 비트가 오유라고 하자. 얼마나 많은 블로크들에 이 오유가 전파되는가? 수신자에게 어떤 영향을 주는가?

13. 8-bit문자 CFB방식에서 암호문의 전송에 어떤 비트오유가 생긴다면 오유는 얼마나 멀리 전파되는가?

부록 3: 벤트함수

이 부록에서는 벤트함수와 S-통에 대한 그것들의 응용을 정의하고 그 의미를 밝혔다.

n-bit옹근수를 한 비트에로 넘기는 함수 f(x)를 고찰하자. 이것을 흔히 $f:\{0,1\}^n \to \{0,1\}$로 표시한다. 인수 x는 nbit $(x_{n-1},\cdots x_1 x_0)$렬이다. n=3일 때 {g(000)=0; g(001)=1; g(010)=1; g(011)=0; g(100)=1; g(101)=0; g(110)=1; g(111)=0;}과 같이 정의할수 있다. 그러한 함수를 렬의 K번째 칸에 g(K)를 대응시키는 렬벡토르로 표현할수 있다. 즉 우의 실례는 다음과 같다.

$$g = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

함수 $f:\{0,1\}^n \to \{0,1\}$의 왈슈(walsh)의 변환은 다음식으로 정의된다.

$$W_f(w) = \sum_{x=0}^{2^n-1} (-1)^{f(x)+w\cdot x}$$

여기서

$$w \cdot x = w_{n-1}x_{n-1} \oplus \ldots \oplus w_0 x_0$$

이며 w는 $0 \le w \le 2^{n-1}$범위의 옹근수이다. 더하기에서 매 항은 +1혹은 -1만을 취한다. 따라서 주어 진 w의 값에 대하여 $W_f(w)$의 값은 $-2^n \le W_f(w) \le 2^n$범위의 옹근수로 된다.

$f(x)$를 왈슈의 거꿀변환을 리용하여 다음과 같이 표시할수 있다.

$$f(x) = \frac{1}{2^n} \sum_{w=0}^{2^n-1} W_f(w)(-1)^{w\cdot x}$$

즉 $f(x)$는 w의 함수들의 합으로서 표시된다.

N이 짝수일 때 벤트함수의 모임은 $f:\{0,1\}^n \to \{0,1\}$의 함수의 모임이다. 여기서

$$W_f(w) = \pm 2^{n/2} \qquad \forall w \in \{0,1\}^n$$

벤트함수의 실제적정의는 약간 추상적이다. 즉 그것은 왈슈변환이 상수값을 가지는 2진벡토르이다.

이것을 단편적으로 고찰하자. 왈슈변환은 푸리예변환과 비슷하지만(기본적인 공학적 개념의 류사성) 련속인 시간에 관계되는 함수가 아니라 리산2진벡토르이다. 푸리예변환과 마찬가지로 왈슈변환은 다른 "령역"에로 이끌어 간다(시간 대 주파수로가 아니라 개념적으로 류사하다). 변환된 벡토르의 절대값을 취하면 물론 모두 정의옹근수를 준다. 만일 이 수들이 다 같다면 그 변환값을 상수(flat) 혹은 상수값(flat magnitude)을 가지는 왈슈변환이라고 말한다.

이것의 의미는 무엇인가? 다시 푸리예변환과 대비하여 고찰하자. 푸리예변환은 변환전의 시간령역의 함수가 변환후에도 주파수령역의 량으로 된다. 왈슈변환에서도 마찬가지이다. 변환된 벡토르는 처음 초기벡토르에 관여되는 쪼각(piece)들의 전부이다. 차이는 왈슈변환에서 수행되는 연산이 어떤 특정의 행렬과의 곱하기라는것이다(왈슈행렬은 기수이면 충분하다). 이 행렬은 필요한 길이를 가지는 모든 선형2진벡토르의 합성인데 변환 그 자체는 초기벡토르의 선형벡토르모임으로의 넘기기이다. 따라서 상수값은 모든 선형벡토르들이 론의하는 벡토르에 똑같이 기여함을 의미한다. 이것은 비선형성을 반영한다.

이것이 S-통에 어떻게 관계되는가? $n\times m$의 S-통은 n-bit입력을 m-bit출력으로 넘긴다. 이미 언급한것처럼 $n\times m$의 S-통은 m개 비트들의 2^n개의 행들로 이루어 진다. 입력의 nbit는 S-통의 행의 하나를 선택하며 그 행의 m개 비트들이 출력된다. 그러므로 S-통을 m개 렬벡토르 ¡$[c_{m-1}(x)...c_1(x)c_0(x)]$의 모임으로 볼수 있다. 즉 S-통의 매렬은 $c_i:\{0,1\}^n\rightarrow\{0,1\}$과 같은 함수로 볼수 있다. 이로부터 렬들이 벤트함수인 S-통의 구조와 리용에 대하여 고찰할수 있다.

벤트함수는 많은 흥미 있는 성질을 가진다. 그중에서 최대의 비선형성과 완전히(최고차에서) 엄격한 사태성기준(Strict Avalanche Criterion)은 S-통설계에서 가장 흥미있는것이다. 이 리상적인 특성은 4장에서 론의되는 CAST설계과정에서 발휘된다.

제4장. 전통암호: 알고리듬

이 장에서는 현재 리용되고 있는 몇가지 대칭블로크암호들에 대하여 취급한다. 이 암호들은 다음과 같은 기준들에 기초하여 선정되였다.

1. 암호들은 상당한 암호강도를 가진다.
2. 암호들은 인터네트상의 응용프로그람들에서 보편적이다.
3. 암호들은 DES가 도입된 이후 개발되여 온 현대 대칭블로크암호기술들을 반영한다.

여기서는 3중DES, IDEA, Blowfish, RC5, CAST, RC2알고리듬들이 고찰된다. 이 장의 마지막에 개략된 대칭블로크암호들의 중요특징에 대하여 개괄하였다.

4.1 3중DES

힘내기공격에 대한 DES의 잠재적인 약점이 발견되면서 그의 대안을 개발하는데 많은 주의가 돌려 졌다. 한가지 방도는 완전히 새로운 알고리듬을 설계하는것인데 그에 대한 여러가지 실례들을 이 장에서 보여 주었다. 쏘프트웨어와 장비측면에서 현존투자를 보존하는 다른 방도는 DES의 다중암호화와 다중열쇠들을 리용하는것이다. 여기서는 먼저 두번째 방도의 가장 단순한 실례를 고찰한다.그 다음 광범히 쓰이고 있는 3중DES 방법을 보여 준다.

2중DES

다중암호화의 가장 단순한 형태는 두개의 암호화단계와 두개의 열쇠들을 리용하는것이다(그림 4-1의 ㄱ). 평문 P와 두개의 암호열쇠 K_1, K_2이 주어 졌을 때 암호문 C는

$$C = E_{K_2}[E_{K_1}[P]]$$

로서 생성된다.

복호화에서는 그 열쇠들을 거꿀순서로 적용한다. 즉

$$P = D_{K_1}[D_{K_2}[C]]$$

DES에 대하여 이 방식은 열쇠길이가 56×2=112bit이므로 암호강도는 증가한다. 그러나 보다 엄격히 검사하여야 한다.

한단계에로의 변환

모든 56-bit열쇠값들에 대하여 확실하다면 임의의 두개의 열쇠 K_1와 K_2에 대하여

$$E_{K_2}[E_{K_1}[P]] = E_{K_3}[P] \quad (4\text{-}1)$$

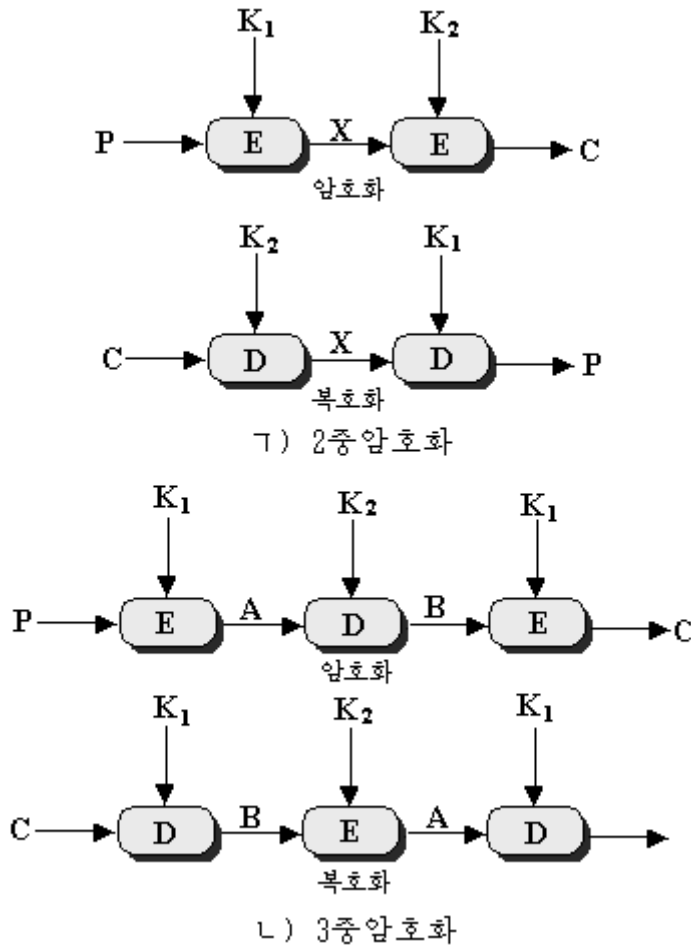



그림 4-1. 다중암호화

를 만족시키는 열쇠 K_3을 얻을수 있다. 이 경우에 2중암호화 및 몇단계로 구성된 DES의 다중암호화는 그 결과가 단순히 56bit열쇠를 리용하는 단순DES와 동등하므로 그리 쓸모가 없다.

이렇게 놓고 볼 때 식 4-1이 꼭 성립한다고는 볼수 없다. DES의 암호화를 64bit블로크의 64bit블로크에로의 넘기기로서 고찰하자. 넘기기는 치환으로 볼수 있다. 즉 가능한 모든 2^{64}개의 입력블로크들을 생각하면 특정한 열쇠에 대한 DES암호화는 매개 블로크들을 유일한 64bit블로크로 넘긴다. 그렇지 않고 주어 진 두개의 입력블로크들이 같은 출력블로크로 넘겨 진다면 본래의 평문을 얻는 복호화는 불가능하다. 2^{64}개의 가능한 입력들에 대하여 입력블로크들의 치환을 생성하는 서로 다른 넘기기들이 얼마나 있는가? 그 값은

$$(2^{64})!=10^{347380000000000000000}>(10^{10^{20}})$$

이라는것을 쉽게 알수 있다. 다른 한편 DES는 매개의 서로 다른 열쇠들에 대하여 하나의 넘기기를 정의한다. 전체 넘기기의 수는

$$2^{56} < 10^{17}$$

이다. 그러므로 만일 DES가 서로 다른 열쇠들로써 두번 적용되였다면 DES의 한번 적용으로는 정의하지 못하는 많은 넘기기들중의 하나를 생성한다고 가정하는것이 지당하다. 이 가정에 대한 많은 증거들이 있었지만 그 가정은 1992년에야 비로소 증명되였다(문헌[CAMP92]).

중간대조공격

즉 2중암호화를 리용하면 단순DES의 암호화와 같지 않은 넘기기가 생성될수 있다. 그러나 이러한 방식을 공격하는 다른 방법이 있는데 그것은 DES의 어떠한 개별적성질에도 의존하지 않으면서 임의의 블로크암호도 공격할수 있다.

중간대조공격법(meet-in-the middle)으로 알려 진 이 알고리듬은 문헌[DIFF77]에서 처음으로 발표되였다. 그것은

$$C = E_{K_2}[E_{K_1}[P]]$$

이라면(그림 4-1의 ㄱ을 보시오.)

$$X = E_{K_1}[P] = D_{K_2}[C]$$

이라는 고찰에 기초하였다. 이미 알려 진 쌍 (P, C)가 주어 졌을 때, 그 공격법은 다음과 같다. 먼저 K_1에 대한 가능한 2^{56}개의 모든 값들에 대하여 P를 암호화한다. 그 결과들을 표에 보관하고 X의 값들에 따라 그 표를 분류한다. 다음 K_2에 대한 2^{56}개의 가능한 모든 값들을 리용하여 C를 복호한다. 매번 복호가 진행될 때마다 그 결과가 표와 일치하는가를 검사한다. 만일 일치되면 얻어 진 두개의 열쇠들을 새로운 기지의 평문-암호문쌍에 대하여 검사한다. 만일 그 두개의 열쇠들이 정확한 암호문을 생성하면 정확한 열쇠라고 인정한다.

어떤 주어 진 평문 P에 대하여 2중DES로써 생성할수 있는 2^{64}개의 가능한 암호문값들이 있다. 2중DES는 실지 112bit열쇠를 리용하므로 가능한 열쇠개수는 2^{112}개이다.

그러므로 평균적으로 볼 때 주어 진 평문 P에 대하여 주어 진 암호문 C를 생성하는 서로 다른 비트열쇠의 개수는 $2^{112}/2^{64} = 2^{48}$개이다. 그러므로 우와 같은 방법은 첫 (P, C)쌍에 대하여 2^{48}개가 실패할것이다. 마찬가지의 론의과정에 추가적인 64bit의 기지평문과 암호문에 대하여 오유통보률은 $2^{48-64} = 2^{-16}$이라는것을 알수 있다. 다시말하여 우와 같은 공격이 알려 진 평문-암호문의 두개 블로크우에서 수행된다면 정확한 열쇠가 얻어 질 확률은 $1-2^{-16}$으로 결정된다. 그 결과는 기지평문공격이 112bit의 열쇠길이를 가지는 2중DES에 대하여 단순DES에서 요구되는 2^{55}오다보다 그닥 좋지 못한 2^{56}오다로서 성공적이라는것을 보여 준다.

두개의 열쇠를 가진 3중 DES

우에서 언급한 공격법을 막기 위한 명백한 대책은 3개의 서로 다른 열쇠들을 리용한 3단계암호화이다. 이것은 기지평문공격비용이 2^{112}로서 현재와 먼 장래의 실천범위를 벗어 나지만 이것은 56×3=168bit의 열쇠길이를 요구한다는 결함이 있으므로 널리 쓰이지 못하고 있다.

그 대안으로서 트츠만(Tuchman)은 두개의 열쇠만을 리용하는 3중암호화방법을 제안하였다[TUCH 79]. 그 함수는 암호-복호-암호(DES)렬에 따른다(그림 4-1의 ㄴ). 즉

$$C = E_{K_1}[D_{K_2}[E_{K_1}[P]]]$$

두번째 단계에서 복호를 리용하는것은 아무런 암호학적의의도 없다. 다만 3중DES의 사용자들이 이전 시기의 단순DES사용자들이 암호화했던 자료를 복호할수 있게 한다. 즉

$$C = E_{K_1}[D_{K_2}[E_{K_1}[P]]] = E_{K_2}[P]$$

두개의 열쇠를 가지는 3중DES는 단순DES에 대한 비교적 보편적인 방법으로서 열쇠관리규격인 ANS X9.17(American National Standard: ANS, 금융열쇠관리협회. 명칭으로 보면 X9.17은 규격이 불명확해 보인다. 이 규격에서 정의된 수많은 기술들은 이 책 전반을 통하여 알수 있는것처럼 다른 표준규격들과 응용에서 널리 리용되여 왔다.)파 ISO 8732를 리용할수 있게 되여 있다.

현재까지는 3중DES에 대한 실천적인 암호분석공격법이 없다. 코퍼스미스(Copper smith)의 문헌[COPP94]에서는 3중DES에서의 전수열쇠탐색방법은 그 비용이 $2^{112}=(5\times10^{33})$오다이라는것을 언급하고 다른 암호분석비용은 단순DES에 비하여 지수적으로 증가하여 10^{52}를 넘는다고 보고 있다.

실천적이 못되지만 보다 앞으로의 성공적인 공격들의 기초를 이루는 여러가지 공격형태들에 대하여 중요한 3중DES에 대한 여러가지 공격방법들을 고찰하는것은 의의가 있다.

첫 제안은 머클(Merkle)과 헬만(Hellman)의 문헌[MERK81]에서 찾아 볼수 있다. 그들의 계획은 A=0(그림 4-1의 ㄴ)인 첫 중간단위값을 생성하는 평문값들을 얻고 그다음 두개의 열쇠를 결정하기 위하여 중간대조공격을 리용하는것이다. 들여야 할 품은 2^{56}이지만 그 기술은 2^{56}개의 선택평문-암호문쌍을 요구한다. 이것은 열쇠소유자가 제공할수 없는 수일것이다.

기지평문공격은 문헌[OORS90]에서 보여 주었다. 이 방법은 선택평문공격법을 개선한것이지만 보다 많은 비용을 요구한다. 그 공격법은 A와 C(그림 4-1의 ㄴ)를 알면 2중DES에 대한 공격으로 귀착된다는데 기초하고 있다. 물론 공격자는 P와 C가 알려 졌어도 두 열쇠를 모르면 A를 알수 없다. 그러나 공격자는 A의 가능한 값을 선택할수 있으며 그다음 A를 생성하는 이미 알려 진 (P, C)쌍을 얻으려고 한다.

그 공격은 다음과 같이 진행된다.

1. n개의 (P, C)쌍을 얻는다. 이것은 이미 알려 진 평문이다. 이것들을 P의 값에 따라 분류하여 표(표 1)에 넣는다(그림 4-2의 ㄴ).

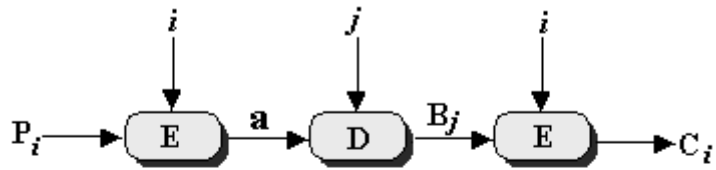



그림 4-2. 3중DES에 대한 기지평문공격

2. A의 임의의 값 a를 잡고 다음과 같은 방법으로 정의된 성원들로 이루어 진 두번째 표(그림 4-2의 ㄷ)를 창조한다. 2^{56}개의 매 가능한 열쇠 $K_1=i$에 대하여 a를 생성하는 평문값 Pi를 계산한다. 즉

$$Pi=Di\,[a]$$

표 1의 성원과 일치하는 매 Pi에 대하여 K_1값을 가정하면서 표 1로부터 (P, C)쌍을 생성하는 B의 값과 K_1값으로 이루어 지는 표 2의 성원들을 창조한다. 즉

$$B=Di\,[C]$$

마지막으로 B의 값에 따라 표 2를 분류한다.

3. 이때 표 2에서 K_1의 여러가지 후보값들을 얻을수 있으며 그것들로써 K_2의 값에 대한 탐색을 할수 있다. 2^{56}개의 매 가능한 열쇠 K_2=j에 대하여 선택한 값 a에 대한 두번째 중간값을 계산한다.

$$B_j=D_j\,[a]$$

매 단계마다 표 2로부터 B_j를 찾는다. 그러한것이 있으면 표 2의 대응하는 열쇠 i와 이 값 j는 미지열쇠(K_1, K_2)들에 대한 후보값이다. 그것은 기지의 (P, C)쌍을 생성하는 열쇠쌍 (i, j)를 발견했기때문이다(그림 4-2의 ㄱ).

4. 몇개의 서로 다른 평문-암호문쌍들에 대하여 매 후보열쇠쌍 (i, j)를 검사한다. 만일 그 열쇠쌍이 문제의 암호문을 생성하면 파업은 끝난다. 그 쌍이 암호문을 생성하지 못하면 새로운 a값을 주어 단계 1로부터 반복한다.

주어 진 기지의 (P, C)에 대하여 열쇠들을 얻는데 성공할수 있는 유일한 값 a를 선택할 확률은 $1/2^{64}$이다. 때문에 n개의 (P, C)쌍에 대하여 값 a가 얻어 질 확률은 $n/2^{64}$이다. 확률론의 기초적인 결과들중 하나로서 n개의 붉은색 공들과 $N-n$개의 풀색 공들이 들어 있는 통에서 하나의 붉은색 공을 꺼내는 회수는 공들이 바뀌여 지지 않는 한 $(N+1)/(n-1)$이다. 이로부터 실행하여야 할 a값들의 기대되는 개수는 n이 클 때 $\frac{2^{64}+1}{n+1} \approx \frac{2^{64}}{n}$로 된다. 따라서 우와 같은 공격법의 실행시간은 $(2^{56})\frac{2^{64}}{n} = 2^{120-\log_2 n}$이다.

3개의 열쇠를 가지는 3중DES

비록 우와 같은 공격법들이 실천적인것 같지 않지만 2열쇠3중DES를 리용하는 사람에게는 흥미가 있을수 있다. 즉 많은 연구사들이 3열쇠3중DES가 더 좋은 후보암호로 될수 있다는것을 느끼기 시작하였다([KALI 96]). 3열쇠3중DES는 168bit의 효과적인 열쇠를 가지는데 다음과 같이 정의된다.

$$C = E_{K_3}[D_{K_2}[E_{K_1}[P]]]$$

K_3=K_2 혹은 K_1=K_2로 놓으면 단순DES의 경우로 된다.

PGP나 S/MIME를 비롯한 수많은 인터네트관련의 응용들에서는 3열쇠3중DES를 리용하는데 이에 대해서는 12장에서 론의한다.

4.2 국제자료암호화알고리듬

국제자료암호화알고리듬(International Data Encryption Algorithm: IDEA)은 스위스련방기술협회(Swiss Federal Institute of Technology)의 쓰웨자 라이(Xuejia Lai)와 제임스 마쎄이(James Massey)가 개발한 대칭블로크암호이다. 초판은 문헌[LAI90]에 제시되였다. 차분암호분석공격에 대처하여 설계된 개량판은 문헌[LAI91]에서 서술하였으며 보다 상세한 내용은 문헌 [LAI92]에서 참고할수 있다.

IDEA는 3장에서 론의한것처럼 자기 생명주기의 마감에 도달한 DES를 대신하여 최근시기에 제안된 많은 전통암호알고리듬들중의 하나이며 그중에서 거의 완성된것으로 인정되고 있다. IDEA는 PGP에서 리용되고 있으며(12장에서 론의) 빠른 속도로 광범히 리용되고 있는 알고리듬이라고 볼수 있다.

설계원리

IDEA는 64bit크기로 블로크화된 자료를 128-bit열쇠를 리용하여 암호화하는 블로크 암호이다. 반면에 DES는 64-bit블로크들을 사용하지만 열쇠길이는 56bit였다. IDEA의 설계목표는 암호강도를 높이는것과 실현을 간단하게 하는것이다.

암호강도

다음과 같은 IDEA의 특성들은 암호강도와 련관된다.

- **블로크길이:** 블로크의 길이는 통계적해석을 할수 없을 정도로 길어야 한다(즉 어떤 블로크들이 다른 블로크들보다 더 자주 나타나는 편향을 없애야 한다). 한편 어떤 효과적인 암호화함수라고 할 때에는 복잡성이 블로크크기에 따라 지수적으로 증가하도록 하여야 한다(문헌[WEGE87]). 64bit블로크크기의 리용은 일반적으로 강한것으로 인정되고 있다. 게다가 연산에 대한 암호반결합(feedback)방식의 리용은 알고리듬의 강도를 이와 같은 측면에서 보다 강하게 한다.
- **열쇠길이:** 열쇠길이는 열쇠탐색을 할수 없을 정도로 길어야 한다. IDEA의 열쇠길이는 128bit로서 이 측면에서 앞으로도 안전할수 있다.
- **혼란(confusion):** 암호문은 평문과 열쇠가 복잡하게 얽히여 생성되여야 한다. 목적은 암호문의 통계적특성이 평문의 통계적특성에 어떻게 관계되는가를 알수 없게 복잡하게 하는것이다. IDEA는 3개의 서로 다른 연산들을 리용하여 그 목적을 달성한다. 이것은 DES와 대조적이며 원리적으로 XOR연산과 작은 비선형 S-통들에 따른다.
- **확산(diffusion):** 매 평문비트들이 모든 암호문비트에 영향을 주어야 하며 매 열쇠비트들도 모든 암호문비트에 영향을 주어야 한다. 많은 암호문비트들에로 하나의 평문비트를 확산하는것은 평문의 통계적구조를 숨길수 있게 한다. IDEA는 이 점에서 매우 효과적이다.

마지막 두가지 점들에 대하여 구체적으로 고찰하자. IDEA에서 혼란은 3개의 서로 다른 연산들을 혼합하여 실현된다. 매개 연산은 하나의 16-bit출력을 얻는데 두개의 16-bit입력을 리용한다. 이 연산들을

- $\oplus$로 표식된 비트별 배타적론리합
- 입력과 출력이 부호 없는 16-bit옹근수로 처리되는 mod 2^{16}(65536)에 관한 옹근수들의 더하기. 이 연산은 $\boxplus$로 표시된다.
- 모두 0인 블로크는 2^{16}으로 취급한것을 제외하고 입출력이 모두 부호 없는 16-bit옹근수로 처리되는 mod 2^{16}+1(65537)에 관한 옹근수곱하기. 이 연산을 $\odot$로 표시한다.

실례로

$0000000000000000 \odot 1000000000000000 = 1000000000000001$

왜냐하면

$$2^{16} \times 2^{15} \bmod (2^{16}+1) = 2^{15}+1$$

이기때문이다.

표 4-1에 2-bit수(16-bit수들로 하기보다)들에 대한 우의 3개 연산결과를 보여 주었다. 이 3개 연산들은 다음과 같은 의미에서 량립되지 않는다.

1. 이 3개 연산들의 어느 쌍도 분배법칙을 만족시키지 못한다. 실례로

$$a \boxplus (b \odot c) \neq (a \boxplus b) \odot (a \boxplus c)$$

표 4-1. IDEA에서 리용된 함수(2bit길이에 대한 연산)

X	Y	X ⊞ Y	X⊙Y	X⊕Y
0 00	0 00	0 00	1 01	0 00
0 00	1 01	1 01	0 00	1 01
0 00	2 10	2 10	3 11	2 10
0 00	3 11	3 11	2 10	3 11
1 01	0 00	1 01	0 00	1 01
1 01	1 01	2 10	1 01	0 00
1 01	2 10	3 11	2 10	3 11
1 01	3 11	0 00	3 11	2 10
2 10	0 00	2 10	3 11	2 10
2 10	1 01	3 11	2 10	3 11
2 10	2 10	0 00	0 00	0 00
2 10	3 11	1 01	1 01	1 01
3 11	0 00	3 11	2 10	3 11
3 11	1 01	0 00	3 11	2 10
3 11	2 10	1 01	1 01	1 01
3 11	3 11	2 10	0 00	0 00

2. 이 3개 연산들은 결합법칙을 만족시키지 못한다. 실례로

$$a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus c$$

이 3가지 서로 다른 연산들을 조합하여 리용하는것은 입력에 복잡한 변형을 하여 단순히 XOR함수만에 의존하는 DES와 같은 알고리듬보다 암호분석을 훨씬 더 어렵게 한다.

IDEA에서 **확산은** 곱하기/더하기구조(MA: Multiplication Addition)로 알려 진 기초적인 알고리듬에 의해 제공된다(그림 4-3). 이 구조는 입력으로서 평문으로부터 얻어지는 두개의 16-bit값들과 열쇠로부터 얻어 지는 두개의 16-bit열쇠값을 취하여 두개의 16-bit출력을 생성한다. 철저한 콤퓨터검사를 진행하여 첫단(first round)의 매 출력비트는

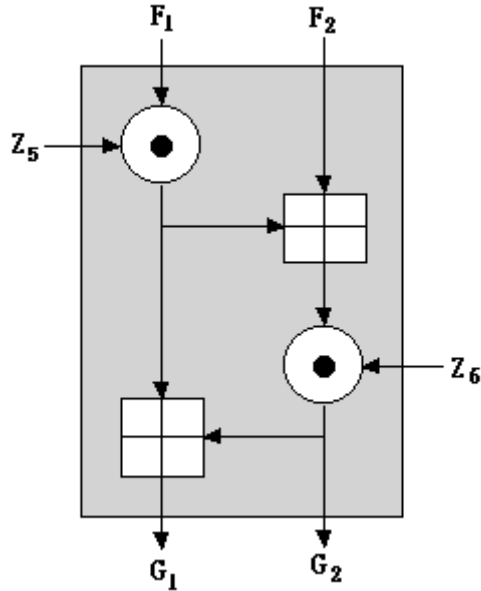



그림 4-3. 곱하기/더하기 (MA)구조

입력으로 주어 지는 평문의 모든 비트와 부분열쇠들의 매 비트에 의존한다는것을 알았다. 이 구조는 알고리듬에서 8번 반복되여 매우 효과적인 확산을 제공한다. 게다가 이 구조가 완전확산을 실현하면서도 최소개수(4개)의 연산들을 리용한다는것을 알수 있다 [LAI91].

실현

IDEA는 쏘프트웨어와 하드웨어실현을 둘 다 쉽게 구성하도록 설계되였다. VLSI에서 전형적인 장치실현은 고속성을 이룩하도록 설계되였다. 쏘프트웨어실현은 유연성과 저가격성이라는 우점을 가진다. 문헌[LAI90]에는 다음과 같은 설계원칙이 서술되였다.

- 쏘프트웨어실현을 위한 설계원칙
  - 부분블로크의 리용: 암호연산은 8, 16 혹은 32bit와 같은 쏘프트웨어에 알맞는 부분블로크들을 처리하도록 되여야 한다. IDEA는 16-bit부분블로크들을 리용한다.
  - 간단한 연산들의 리용: 암호화연산들은 더하기, 밀기 등을 리용하여 쉽게 프로그람화될수 있어야 한다. IDEA의 3가지 기초연산들은 이 요구를 만족시킨다. 3가지 연산들중 가장 어려운 mod (2^{16}+1)의 곱하기는 실제로는 간단한 기초연산들로부터 얻어 진다(문제 4-4를 보시오).
- 장치실현을 위한 설계원칙
  - 암호화 및 복호화의 단순성: 암호화와 복호화는 다만 열쇠를 리용하는 방

법에서만 차이나고 같은 장치가 두 부분에서 다 쓰이도록 구성하여야 한다. DES와 마찬가지로 IDEA도 이 요구를 만족시키는 구조를 가진다.

- 규칙적인 구조: 암호에서 VLSI실현을 간단하게 하기 위해서는 규칙적인 모듈구조를 가져야 한다. IDEA는 여러번 반복되는 두개의 기초모듈블로크들로 구성되였다.

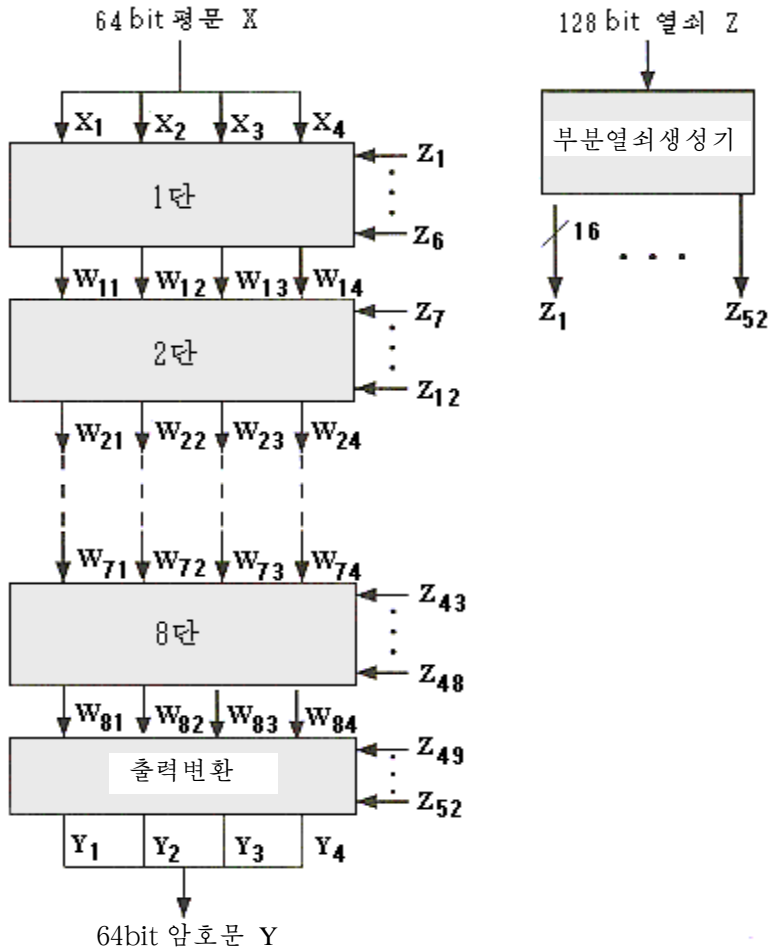



그림 4-4. IDEA의 총 구조도

IDEA 암호화

IDEA의 총적방식은 그림 4-4와 같다. 모든 암호방식들과 마찬가지로 암호화함수에는 두개의 입력 즉 평문과 열쇠가 있다. 이 경우 평문은 64bit길이이며 열쇠는 128bit길이이다. 그림의 왼쪽켠에서 알수 있는것처럼 IDEA는 마지막변환함수를 제외하고는 8개 단으로 되여 있다. 알고리듬은 입력을 4개의 16-bit부분블로크들로 분할한

다. 매단은 입력으로서 4개의 16-bit부분블로크를 가지고 4개의 출력부분블로크를 내보낸다. 마지막변환도 4개의 16-bit부분블로크를 내보내며 이것들은 묶어 져 64-bit의 암호문을 생성한다. 매단은 또한 6개의 16-bit부분열쇠들을 리용하며 마지막변환이 4개의 부분열쇠들을 리용함으로써 총 52개의 16-bit부분열쇠들을 리용하게 된다. 그림 4-4의 오른쪽 그림은 이 52개의 부분열쇠들이 모두 초기의 128-bit열쇠로부터 생성된다는것을 보여 준다.

한개 단의 세부

그림 4-5에서 보여 준것처럼 알고리듬의 한개 단에 대하여 보다 구체적으로 고찰하자. 사실 그림 4-5는 IDEA의 첫단이다. 그밖의 단들은 같은 구조를 가지지만 입력과 부분열쇠들이 다르다. 그림을 보고 알수 있는것처럼 IDEA는 고전페이스텔구조가 아니다. 이 단은 더하기와 곱하기연산을 리용하여 4개의 입력부분블로크들을 4개의 부분열쇠들과 결합하는 변환으로부터 시작한다. 이 변환의 4개의 출력부분블로크들을 결합하고 XOR연산을 리용하여 MA구조의 입력인 두개의 16-bit블로크들을 생성한다. 구조는 입력으로서 두개의 부분열쇠를 가지며 그 입력들을 조합하여 두개의 16-bit출력을 생성한다.

마지막으로 우의 변환의 4개의 출력블로크와 MA구조의 두개의 출력블로크를 XOR로 결합하여 이 단의 4개의 출력블로크를 형성한다. 두번째와 세번째 입력(X_2과 X_3)들에 의하여 부분적으로 생성된 두 출력들은 서로 바뀌여 두번째와 세번째 출력(W_{12}과 W_{13})들을 생성한다. 이것은 처리가 진행됨에 따라 비트들의 혼란을 증가시켜 차분암호분석공격을 억제하도록 한다.

그림 4-4에서 출력변환단으로 표시된 알고리듬의 아홉번째 단계는 그림 4-6에 보여주었다. 이것은 이전 단들(그림 4-5)의 MA구조 웃부분에 있는 어두운 4각형과 같은 구조를 가진다. 다만 차이점은 두번째와 세번째 입력들이 연산에 참가하기전에 서로 바뀐다는 점이다. 사실 이것은 8번째 단의 끝에서 서로 바꿈을 진행하지 않은것과 같은 효과를 가진다. 이러한 자리바꿈을 하는 리유는 복호화와 암호화가 같은 구조를 가지도록 하기 위해서이다. 또한 9번째 단은 4개의 부분열쇠들을 요구하는데 이것은 우의 8개 단들에서 6개의 부분열쇠들을 리용하는것과 차이난다.

부분열쇠생성

그림 4-4를 다시 고찰하자. 52개의 16-bit부분열쇠들이 128-bit암호열쇠로부터 생성된다는것을 알수 있다. 그 생성방식은 다음과 같다. Z_1, Z_2,⋯, Z_8 의 부분열쇠들은 Z_1은 제일웃자리의 16bit이고 Z_2은 다음의 16bit에 대응시켜 나가는 방식의 열쇠로부터 직접 얻어 진다.

그다음 25번의 왼쪽순환밀기가 진행된후 다음 8개의 부분열쇠들이 추출된다. 이 과정을 52개의 부분열쇠들이 얻어 질 때까지 반복한다. 그림 4-7에 기본열쇠에 대한 모든 부분열쇠들의 비트할당을 보여 주었다.

이 방식은 8개 단에서 부분열쇠들로서 리용된 열쇠비트들을 바꾸는 효과적인 기술을 제공한다. 매단에서 리용하는 첫 부분열쇠로는 열쇠비트들의 각이한 모임이 리용된다는것을 지적한다. 만일 열쇠가 Z[1..128]로서 표시되였다면 8개 단의 매개 첫 열쇠는 다음과 같이 비트할당된다.

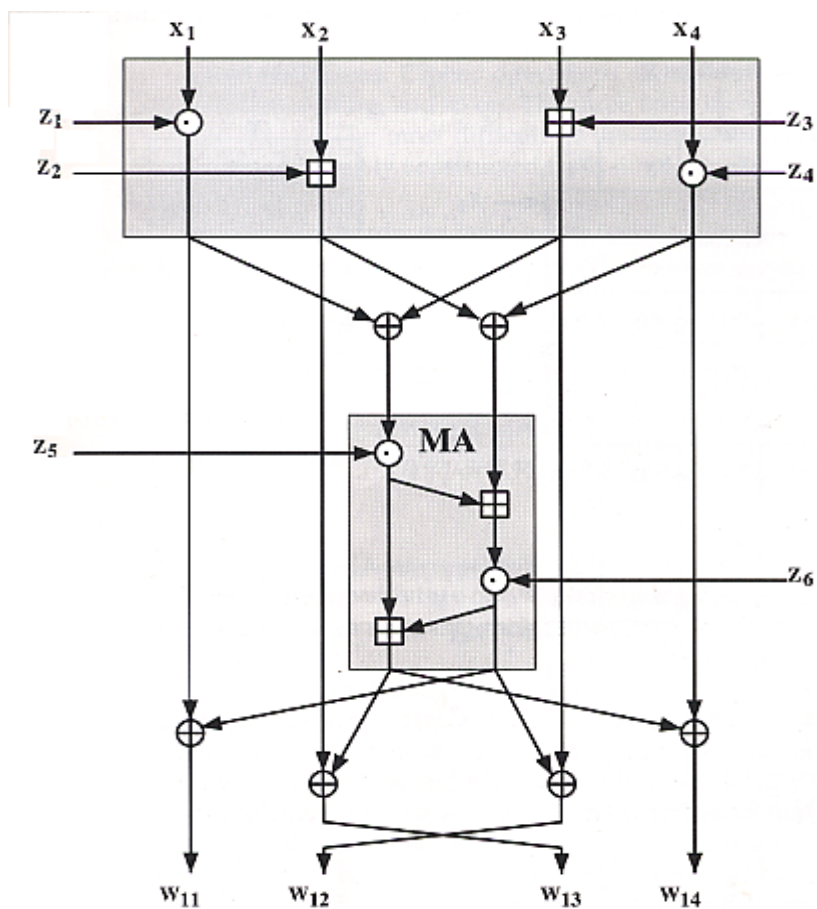



그림 4-5. IDEA의 한개 단(첫단)

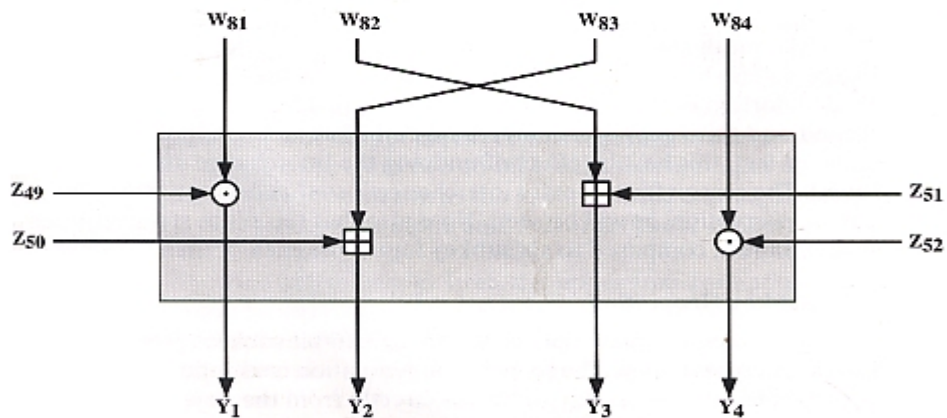



그림 4-6. IDEA의 출력변환단계

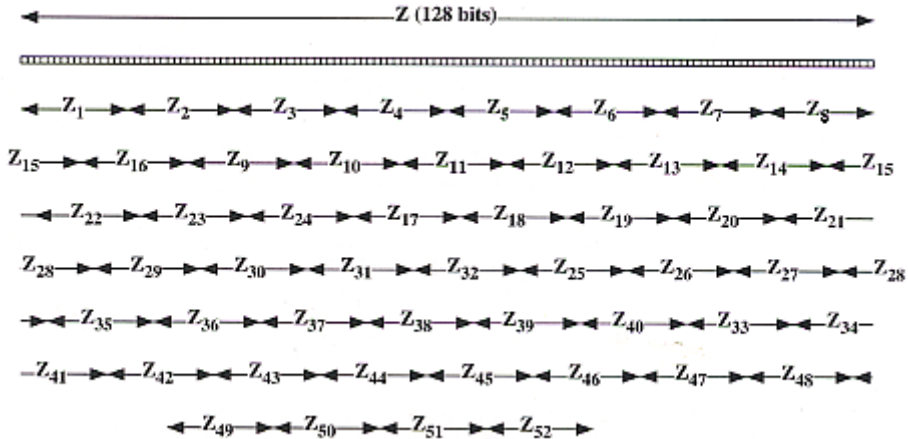



그림 4-7. IDEA부분열쇠

Z_1=Z[1..16]	Z_{25}=Z[76..91]
Z_7=Z[94..112]	Z_{31}=Z[44..59]
Z_{13}=Z[90..105]	Z_{37}=Z[34..52]
Z_{19}=Z[83..98]	Z_{43}=Z[30..45]

첫째 단과 여덟번째 단을 제외하고 매단에서 리용되는 96개 부분열쇠비트들은 련속되여 있지 않으므로 한 단과 다른 단에 있는 부분열쇠들사이에는 간단한 밀기관계조차 없다. 그 리유는 매단에서 6개 부분열쇠들만이 리용되고 8개 부분열쇠들은 그 열쇠의 매번의 순환으로써 얻어 지기때문이다.

IDEA의 복호화

복호화는 암호화와 같다. 복호화는 그림 4-4에서 보여 준 총적인 IDEA구조에 대한 입력으로서 암호문을 리용하여 진행되지만 부분열쇠들은 서로 다르게 선택된다. 복호부분열쇠 $U_1, \cdots, U_{52}$는 다음과 같이 암호화부분열쇠들로부터 얻어 진다.

1. 복호화의 i번째 단의 첫 4개 부분열쇠들은 암호화의 $(10-i)$번째 단의 첫 4개의 부분열쇠들로부터 얻어 진다. 여기서 변환단계는 9개 단으로 된다. 네번째의 복호화부분열쇠들은 대응하는 첫번째와 네번째의 암호화부분열쇠들의 적mod$(2^{16}+1)$에 관한 거꿀과 같다. 2단부터 8단까지의 두번째와 세번째 복호화부분열쇠들은 세번째와 두번째 암호화부분열쇠들의 합mod$(2^{16}+1)$에 관한 반대원소와 같다. 1단과 9단에서 두번째와 세번째 복호부분열쇠들은 대응한 두번째와 세번째 암호화부분열쇠들의 mod 2^{16}에 관한 반대원소와 같다.
2. 첫 8개 단들에 대하여 복호화의 i번째 단의 마지막 두개 복호화부분열쇠들은 $(9-i)$번째 단의 마지막 두개 암호화부분열쇠들과 같다.

표 4－2 암호화와 복호화부분열쇠들

암호화			복호화	
단계	설계	갈음	설계	갈음
단1	$Z_1Z_2Z_3Z_4Z_5Z_6$	$Z[1..96]$	$U_1U_2U_3U_4U_5U_6$	$Z_{49}^{-1}-Z_{50}-Z_{51}Z_{52}^{-1}Z_{47}Z_{48}$
단2	$Z_7Z_8Z_9Z_{10}Z_{11}Z_{12}$	$Z[97..128;26..89]$	$U_7U_8U_9U_{10}U_{11}U_{12}$	$Z_{43}^{-1}-Z_{45}-Z_{44}Z_{46}^{-1}Z_{41}Z_{42}$
단3	$Z_{13}Z_{14}Z_{15}Z_{16}Z_{17}Z_{18}$	$Z[90..128;1..25, 51..82]$	$U_{13}U_{14}U_{15}U_{16}U_{17}U_{18}$	$Z_{37}^{-1}-Z_{39}-Z_{38}Z_{40}^{-1}Z_{35}Z_{36}$
단4	$Z_{19}Z_{20}Z_{21}Z_{22}Z_{23}Z_{24}$	$Z[83..128;1..50]$	$U_{19}U_{20}U_{21}U_{22}U_{23}U_{24}$	$Z_{31}^{-1}-Z_{33}-Z_{32}Z_{34}^{-1}Z_{29}Z_{30}$
단5	$Z_{25}Z_{26}Z_{27}Z_{28}Z_{29}Z_{30}$	$Z[76..128;1..43]$	$U_{25}U_{26}U_{27}U_{28}U_{29}U_{30}$	$Z_{25}^{-1}-Z_{27}-Z_{26}Z_{28}^{-1}Z_{23}Z_{24}$
단6	$Z_{31}Z_{32}Z_{33}Z_{34}Z_{35}Z_{36}$	$Z[44..75;101..128;1..36]$	$U_{31}U_{32}U_{33}U_{34}U_{35}U_{36}$	$Z_{19}^{-1}-Z_{21}-Z_{20}Z_{22}^{-1}Z_{17}Z_{18}$
단7	$Z_{37}Z_{38}Z_{39}Z_{40}Z_{41}Z_{42}$	$Z[37..100;126..128;1..29]$	$U_{37}U_{38}U_{39}U_{40}U_{41}U_{42}$	$Z_{13}^{-1}-Z_{15}-Z_{14}Z_{16}^{-1}Z_{11}Z_{12}$
단8	$Z_{43}Z_{44}Z_{45}Z_{46}Z_{47}Z_{48}$	$Z[30..125]$	$U_{43}U_{44}U_{45}U_{46}U_{47}U_{48}$	$Z_7^{-1}-Z_9-Z_8Z_{10}^{-1}Z_5Z_6$
변환	$Z_{49}Z_{50}Z_{51}Z_{52}$	$Z[23..86]$	$U_{49}U_{50}U_{51}U_{52}$	$Z_1^{-1}-Z_2-Z_3Z_4^{-1}$

그림 4-8. IDEA의 암호화 및 복호화

표 4-2에 이 관계들이 종합되여 있다. Z_j^{-1}은 모드에 관한 곱하기거꿀을 의미하며 따라서 $Z_j \odot Z_j^{-1} = 1$이다. $2^{16}+1$이 씨수이므로 매개 령 아닌 옹근수 $Z_j \le 2^{16}$은 7장에서 보여 주는바와 같이 유일한 거꿀원소를 가진다. $-Z_j$는 mod 2^{16}에 관하여 반대원소를 의미하며 따라서 $-Z_j \boxplus Z_j = 0$을 나타낸다.

그림 4-8에 의하여 이 알고리듬이 복호화부분열쇠들에 대하여 정확한 결과를 준다는 것을 알수 있다. 그림의 왼쪽은 아래방향으로 내려 가는 암호화과정을 보여 주며 오른쪽은 웃쪽으로 올라 가는 복호화과정을 보여 준다. 8개 단들의 매개는 변환과 부분암호화라고 하는 두개의 단계로 또 갈라 지며 그 변환의 부분단계는 그림 4-5의 우에 있는 어

두운 4각형에 대응하고 부분암호화단계는 그 단처리의 나머지에 해당한다.

그림 4-8에 주어 진 두 도식에서 밑의 4각형들을 고찰하자. 암호화쪽에서 다음과 같은 관계들이 출력변환에 대하여 성립한다.

$$Y_1=W_{81} \odot Z_{49} \qquad Y_3=W_{82} \boxplus Z_{51}$$
$$Y_2=W_{83} \boxplus Z_{50} \qquad Y_4=W_{84} \odot Z_{52}$$

복호화처리의 첫단의 첫 부분단계는 다음과 같은 관계들을 만족시킨다.

$$J_{11}=Y_1 \odot U_1 \qquad J_{13}=Y_3 \boxplus U_3$$
$$J_{12}=Y_2 \boxplus U_2 \qquad J_{14}=Y_4 \odot U_4$$

대입하면 다음과 같다.

$$J_{11}=Y_1 \odot Z_{49}^{-1}= W_{81} \odot Z_{49} \odot Z_{49}^{-1}=W_{81}$$
$$J_{12}=Y_2 \boxplus -Z_{50}= W_{83} \boxplus Z_{50} \boxplus -Z_{50}=W_{83}$$
$$J_{13}=Y_3 \boxplus -Z_{51} =W_{82} \boxplus Z_{51} \boxplus -Z_{51}=W_{82}$$
$$J_{14}=Y_4 \odot Z_{52}^{-1}= W_{84} \odot Z_{52} \odot Z_{52}^{-1}=W_{84}$$

이렇게 복호처리의 첫 부분단계의 출력은 두번째와 세번째 블로크들의 내부교체를 제외하고는 암호화처리의 마지막단계의 입력과 같다. 이때 그림 4-5로부터 유도할수 있는 다음과 같은 관계를 고찰하자.

$$W_{81} = I_{81} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$
$$W_{82} = I_{83} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$
$$W_{83} = I_{82} \oplus MA_L(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$
$$W_{84} = I_{84} \oplus MA_L(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})$$

여기서 $MA_R(X, Y)$는 입력 X와 Y를 가지는 MA구조(그림 4-3)의 오른쪽 출력이고 $MA_L(X, Y)$는 입력 X, Y에 대한 MA구조의 왼쪽 출력이다. 이때 다음과 같이 된다.

$$\begin{aligned}
V_{11} &= J_{11} \oplus MA_R(J_{11} \oplus J_{13}, J_{12} \oplus J_{14}) \\
&= W_{81} \oplus MA_R(W_{81} \oplus W_{82}, W_{83} \oplus W_{84}) \\
&= I_{81} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84}) \oplus \\
&\quad MA_R[I_{81} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84}) \oplus I_{83} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84}), \\
&\quad I_{82} \oplus MA_L(I_{81} \oplus I_{83}, I_{82} \oplus I_{84}) \oplus I_{84} \oplus MA_L(I_{81} \oplus I_{83}, I_{82} \oplus I_{84})] \\
&= I_{81} \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84}) \oplus MA_R(I_{81} \oplus I_{83}, I_{82} \oplus I_{84}) \\
&= I_{81}
\end{aligned}$$

마찬가지로

$$V_{12} = I_{83}$$
$$V_{13} = I_{82}$$
$$V_{14} = I_{84}$$

이다.

이처럼 복호화처리의 두번째 부분단계의 출력은 두번째와 세번째 블로크들의 교체를 제외하고는 암호화처리의 마지막부터 다음 부분단계까지의 입력과 같다. 우와 같은 론의의 련속으로 이 관계는

$$V_{81} = I_{11}$$
$$V_{82} = I_{13}$$
$$V_{83} = I_{12}$$
$$V_{84} = I_{14}$$

으로서 그림 4-8의 매개 대응하는 점에서 성립한다는것을 알수 있다.

마지막으로 복호화처리의 출력변환은 두번째와 세번째 블로크들의 교체를 제외하고는 암호화처리의 첫 부분단계와 같으므로 전체 암호화처리의 출력은 그 암호화처리의 입력과 같다는것을 알수 있다.

4.3 BLOWFISH

Blowfish는 브루스 슈나이어(Bruce Schneier)(문헌[SCHN93, SCHN94])에 의하여 개발된 대칭블로크암호이다. Blowfish는 다음과 같은 특성을 가지도록 설계되였다.

- **고속성:** Blowfish는 바이트당 18박자주기의 속도로 32-bit극소형처리기에서 자료를 암호화한다.
- **밀집성:** Blowfish는 5K정도의 기억기에서 동작할수 있다.
- **단순성:** Blowfish의 간단한 구조는 실현하기 쉽고 알고리듬강도를 결정하기가 쉽게 되여 있다.
- **가변적인 안전성:** 열쇠길이는 가변적이고 448bit길이까지 확장할수 있다. 이것은 고속성과 높은 안전성사이에서 균형을 보장한다.

Blowfish는 64bit길이의 평문을 64bit길이의 암호문으로 암호화한다. Blowfish는 여러가지 제품들로 실현되였으며 충분히 검토되였다. 그러므로 Blowfish의 안전성에 대한 문제는 크게 제기되지 않고 있다.

부분열쇠와 S-통의 생성

Blowfish는 32부터 448bit(1~14개의 32bit단어)까지의 범위에 놓이는 열쇠를 리용한다. 열쇠는 18개의 32-bit부분열쇠들과 총 1024개의 32-bit입력들을 포함하는 4개의 8×32 S-통들을 생성하는데 리용된다.

열쇠들은 K-배렬(K_1, K_2, ⋯, K_j, $1 \le j \le 14$)에 보관된다.
부분열쇠들은 P-배렬(P_1, P_2, ⋯, P_{18})에 보관된다.
4개의 S-통들이 있는데 그 매개는 256개의 32bit성분들을 가진다. 즉

$$S_{1,0}, S_{1,1}, ..., S_{1,255}$$
$$S_{2,0}, S_{2,1}, ..., S_{2,255}$$
$$S_{3,0}, S_{3,1}, ..., S_{3,255}$$
$$S_{4,0}, S_{4,1}, ..., S_{4,255}$$

P-배렬과 S-통들을 생성하는 단계는 다음과 같다.

1. 먼저 P-배렬을 초기화하고 다음 4개의 S-통들을 일정한 상수 π의 소수부분의 비트들을 리용하여 차례로 초기화한다. 따라서 π의 매 비트들중 가장 왼쪽에 있는 32bit부터 시작하여 P_1, P_2 등으로 설정된다. 실례로 16진수로 다음식이 성립한다.

$$P_1 = 243F6A88$$
$$P_2 = 85A308D3$$
$$\cdots$$
$$S_{4,254} = 578FDFE3$$
$$S_{4,255} = 3AC372E6$$

2. K-배렬의 단어들을 다시 리용하여 P-배렬과 K-배렬을 비트별로 XOR한다. 실례로 최대길이열쇠(14개의 32-bit단어들)들에 대하여 다음식이 성립한다.

$$P_1 = P_1 \oplus K_1, P_2 = P_2 \oplus K_2, ..., P_{14} = P_{14} \oplus K_{14}, P_{15} = P_{15} \oplus K_1, ...P_{18} = P_{18} \oplus K_4.$$

3. 현재의 P,S-배렬들을 리용하여 모두가 0인 64-bit블로크를 암호화하고 P_1와 P_2을 암호화의 출력으로 교체한다.
4. 현재의 P-배렬, S-배렬들을 리용하여 단계 3의 출력을 암호화하고 그 결과에서 P_3과 P_4을 교체한다.
5. 이 처리를 련속적으로 변하는 Blowfish알고리듬의 출력을 매단에서 리용하여 P의 모든 원소들과 그다음 S의 모든 원소들이 모두 갱신될 때까지 계속한다. 그것을 종합하면 다음과 같다.

$$\begin{aligned} P_1, P_2 &= E_{P,S}[0] \\ P_3, P_4 &= E_{P,S}[P_1 \| P_2] \\ &\cdots \\ P_{17}, P_{18} &= E_{P,S}[P_{15} \| P_{16}] \\ S_{1,0}, S_{1,1} &= E_{P,S}[P_{17} \| P_{18}] \\ &\cdots \\ S_{4,254}, S_{4,255} &= E_{P,S}[S_{4,252} \| S_{4,253}] \end{aligned}$$

여기서 $E_{P,S}[Y]$는 배렬 P와 S의 Blowfish를 리용하여 Y를 암호화할 때 생성된 암호문이다.

Blowfish알고리듬은 모두 521회의 실행으로 S-배렬과 P-배렬을 생성한다. 따라서 이 방법은 비밀열쇠를 주기적으로 바꾸는 응용에 대해서는 적당하지 못하다. 게다가 P-배렬과 S-배렬들은 열쇠생성에서 매번 리용되므로 기억되여야 한다. 이것은 4KB의 기억용량을 요구한다. 때문에 Blowfish는 스마트(smart)카드와 같은 제한된 기억기를 가지는 응용에 대해서는 적용되지 못한다.

암호화와 복호화

Blowfish는 두개의 원시연산들을 리용한다:

- **더하기:** +로 표시되는 단어들의 더하기는 mod 2^3로 수행된다.
- **배타적론리합:** 이 연산은 ⊕로 표시된다.

이 두가지 연산에서 중요한것은 그것들이 교체되지 않는다는것이다. 이것은 암호분석을 보다 어렵게 한다.

그림 4-9의 ㄱ은 암호화연산을 보여 준다. 평문은 두개의 32-bit쌍 LE_0과 RE_0으로

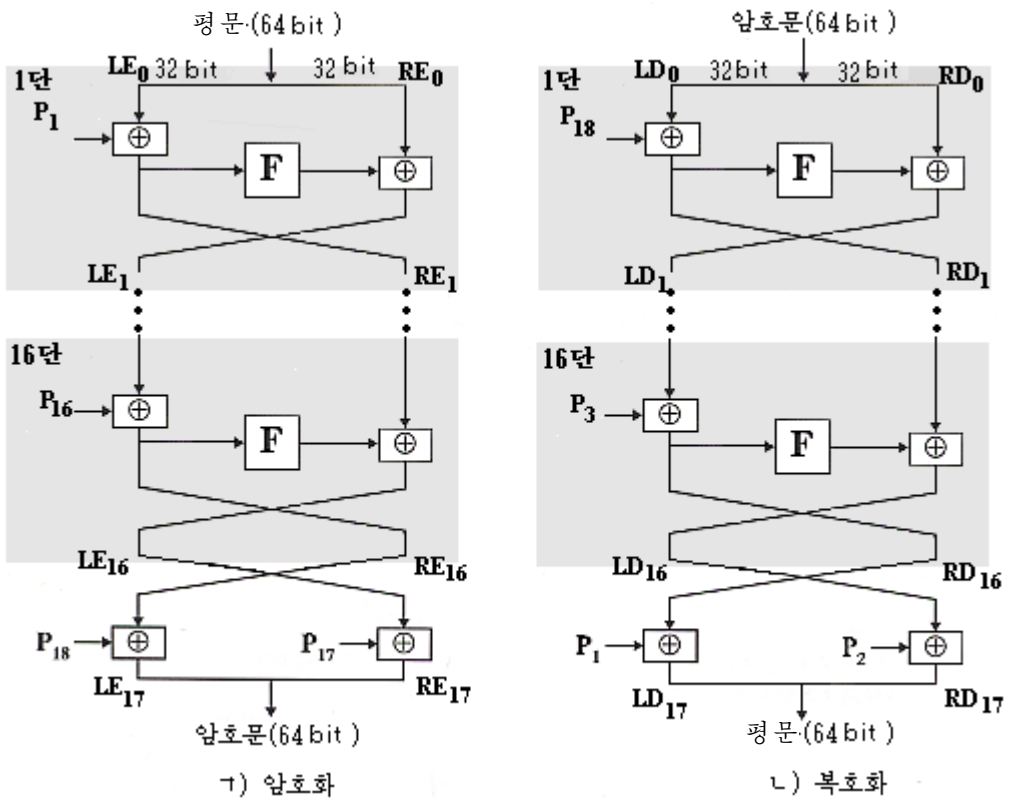



그림 4-9. Blowfish의 암호화와 복호화

나누어 진다. i번째 단이 수행된후 자료를 왼쪽과 오른쪽으로 절반씩 나누는데 LE_i와 RL_i변수들을 리용한다.

알고리듬은 다음과 같은 가상부호로 정의된다.

$$
\begin{aligned}
&\text{for } i=1 \text{ to } 16 \text{ do}\\
&\qquad RE_i = LE_{i-1} \oplus P_i\ ;\\
&\qquad LE_i = F[RE_i] \oplus RE_{i-1}\ ;\\
&LE_{17} = RE_{16} \oplus P_{18}\ ;\\
&RE_{17} = LE_{16} \oplus P_{17}\ ;
\end{aligned}
$$

결과 암호문은 두개의 변수 LE_{17}과 RE_{17}에 포함된다. 함수 F를 그림 4-10에 보여주었다. F에 대한 32bit입력은 4byte들로 나뉘여 진다. 이 바이트들을 a, b, c, d라고 표시하면 함수 F는 다음과 같이 정의된다.

$$F[a,b,c,d] = ((S_{1,a} + S_{2,b}) \oplus S_{3,c}) + S_{4,d}$$

이렇게 매단에서는 mod 2^{32}와 XOR, S-통에 의한 전치가 다 리용된다.

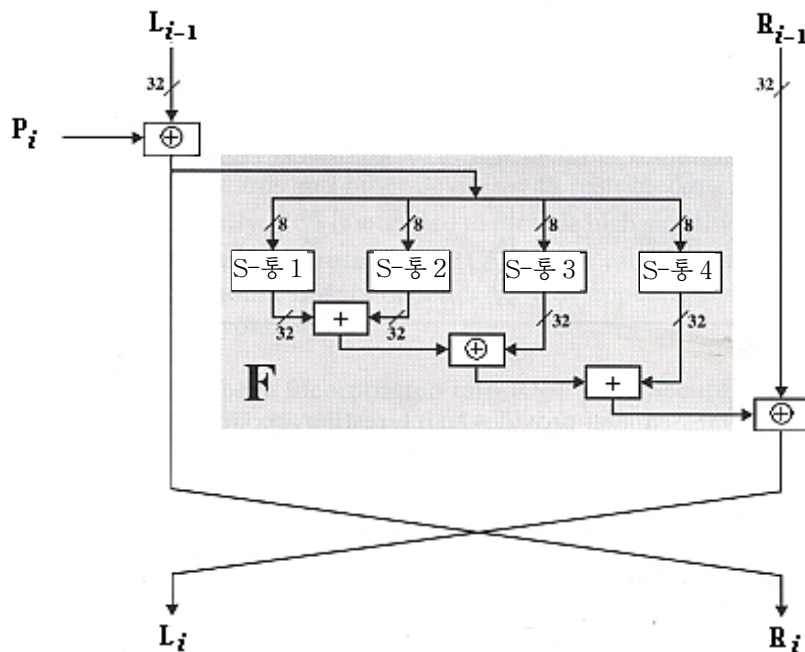



그림 4-10. 하나의 Blowfish단의 세부

그림 4-9의 ㄴ에서 보는바와 같이 복호화는 암호화알고리듬으로부터 쉽게 얻어 진다. 복호화인 경우 64bit의 암호문은 초기 두개의 한단어변수들인 LD_0과 RD_0에 보관된다. 변수 LD_i와 RD_i를 리용하여 i번째 단후에 자료의 왼쪽과 오른쪽 절반을 표시한다. 대부분의 블로크암호들과 마찬가지로 Blowfish복호화도 부분열쇠들을 거꿀순서로 리용한다. 그러나 다른 블로크암호들과 달리 Blowfish복호화는 암호화와 같은 알고리듬방향으로 진행한다. 그 알고리듬은 다음과 같이 정의된다.

$$
\begin{aligned}
&\text{for } i=1 \text{ to } 16 \text{ do} \\
&\qquad RD_i = LD_{i-1} \oplus P_{19-i}\ ; \\
&\qquad LD_i = F[RD_i] \oplus RD_{i-1}\ ; \\
&LD_{17} = RD_{16} \oplus P_1\ ; \\
&RD_{17} = LD_{16} \oplus P_2\ ;
\end{aligned}
$$

론의

Blowfish는 아마도 이 책에서 서술한것들중에서 가장 강한 전통암호알고리듬일것이다. DES와 달리 Blowfish의 S-통들은 열쇠의존형이다. 그러나 RC5와 같은 다른 일부 알고리듬들은 매단에서 수행되는 함수들중 하나가 자료에 의존하도록 설계되였다(RC5인 경우에는 회전). 그러나 Blowfish인 경우에는 부분열쇠들과 S-통들이 Blowfish자체의 반복처리에 의하여 생성된다. 이것은 비트들을 전혀 리해할수 없도록 하며 암호분석을 불가능하게 한다. 지금까지 Blowfish에 대한 몇가지 분석법들이 제안되였는데 실천적인 약점들은 발견되지 않았다.

Blowfish의 또 다른 흥미 있는 측면은 연산들이 고전페이스텔(Feistel)암호와 같이 매단에서 자료의 절반에 대한 연산을 수행하는것이 아니라 자료의 두개의 절반부분들에서 진행한다는것이다. 이것은 더하기연산이 선형(XOR)이라고 해도 더 큰 암호강도를 제공한다. 이에 대해 문헌[HEYS95]에서는 대입치환망(SPN)의 매단에서 선형변환을 포함하는것은 블로크암호의 사태특성을 더 높인다는것이 지적되였다(저자는 Blowfish에 대하여서는 고찰하지 못하고 일반적으로 SNP들을 해석하였다). 힘내기공격법에 대하여 Blowfish는 실제상 448bit만큼 긴 적당한 열쇠길이선택으로 해서 약하지 않다.

Blowfish는 또한 실행이 매우 빠르다. 슈나이어가 제시한 표 4-3은 C언어로 작성한 여러 암호프로그람들의 박자수를 펜티움급에서 비교하였다. 확실히 Blowfish는 고속이였다.

표 4-3. 펜티움에서 블로크암호들의 속도비교

알고리듬	단당 박자 주기	단의 #	암호화된 바이트당 박자주기의 #
Blowfish	9	16	18
RC5	12	16	23
DES	18	16	45
IDEA	50	8	50
3중DES	18	48	108

문헌[SCHN93]에서 슈나이어는 Blowfish를 보다 세부화한 흥미 있는 설계지침을 제시하였다. 그 지침들중 특히 강조할것은 다음과 같은것들이다.

1. 힘내기공격은 부분열쇠생성처리시간으로 하여 아주 어렵다. 하나의 열쇠검사에 요구되는 암호화알고리듬의 총 실행시간은 522이다.
2. 함수 F는 Blowfish로 하여금 페이스텔(Feistel)망에 대하여 아주 훌륭한 련쇄효과를 나타내도록 한다. 즉 i번째 단에서 L_{i-1}의 모든 비트들은 R_{i-1}의 모든 비트들에 영향을 준다. 그리고 매 부분열쇠비트들은 모든 열쇠비트들에서 영향을 받게 되며 따라서 F는 매단이 수행된후 열쇠(P_i)와 자료의 오른쪽 절반(R_i)사이에 완전한 련쇄효과를 나타낸다.
3. F의 매 입력비트들은 하나의 S-통의 입력으로서만 리용된다. 반대로 DES에서는 많은 비트들이 두개의 S-통들에 대한 입력으로 리용되는데 이것은 차분공격법에 대항하여 알고리듬을 보다 강하게 한다. 슈나이어는 이러한 부가된 복잡성이 열쇠의존 S-통들에 대해서는 필요 없다고 보고 있다.
4. CAST와 달리 Blowfish의 F함수는 단수의존형이 아니다. 슈나이어는 P-배렬대입이 단수에 의존한다면 그러한 의존성은 어떠한 암호학적인 우점도 주지 못한다고 보고 있다.

4.4 RC5

RC5는 론 리베스트(Ron Rivest)(문헌[RIVE94, RIVE95])에 의하여 개발된 대칭 블로크암호알고리듬이다.

RC5는 다음과 같은 특성을 가지도록 설계되였다.

- **하드웨어나 쏘프트웨어에 대한 적합성:** RC5는 극소형처리기들에서 공통으로 쓰이는 기초연산들만을 리용한다.
- **고속성:** 이를 위하여 RC5는 알고리듬이 간단하며 대상지향적이다. 다시말하여 기초연산들은 한번에 처리할수 있는 자료단위로 단어를 설정하였다.
- **서로 다른 단어길이들에 대한 처리소자들의 적응성:** 단어의 비트개수는 RC5의 파라메터이다. 즉 서로 다른 단어길이들이 서로 다른 알고리듬들을 내놓는다.
- **가변인 단수:** 단수는 RC5의 두번째 파라메터이다. 이 파라메터는 고속성과 높은 안전성을 담보한다.
- **가변길이열쇠 :** 열쇠길이는 RC5의 세번째 파라메터이다. 이것은 속도와 안전성측면을 높이도록 한다.
- **단순성:** RC5의 단순구조는 도구작성을 실행하기 쉽고 알고리듬의 강도고찰을 간단하게 한다.
- **저기억요구:** 이것은 RC5가 스마트카드들이나 기억기가 제한된 다른 장치들에 적용될수 있도록 한다.
- **강한 보안:** RC5는 편리한 파라메터들의 설정으로 하여 강한 보안을 보장한다.
- **자료의존회전:** RC5는 전체적으로 자료에 의존하는 회전들을(순환 비트밀기) 섞어 구성되였다. 이것은 암호분석에 대항하여 알고리듬의 강도를 높이도록 한다.

RC5는 BSAFE, JSAFE와 S/MAIL을 비롯하여 RSA자료보안회사(RSA Data Security, Inc.)의 주요제품들에 적용되였다.

RC5의 파라메터

RC5에는 다음과 같은 3개의 파라메터들이 있다.

파라메터	정의	허용값
w	단어의 비트크기. RC5는 2단어블로크들을 암호화한다.	16, 32, 64
r	단의 개수	0, 1,…,255
b	비밀열쇠 K의 8bit바이트(옥테드)들의 개수	0, 1, …,255

이처럼 RC5는 32, 64 혹은 128bit길이의 평문블로크들을 같은 길이의 암호문블로크들로 암호화한다. 열쇠길이는 0～2040bit길이 범위에 놓인다. RC5의 개별적변종들은 RC5-$w/r/b$로 표시된다. 실례로 RC5-32/12/16은 암호화와 복호화에 32-bit단어들(64-bit평문과 암호문블로크들), 12단, 16byte의 열쇠길이(128bit)를 가진다. 리베스트(Rivest)는 표준판으로서 RC5-32/12/16을 제기하였다.

열쇠확장

RC5는 비밀열쇠에 대한 연산들의 복합모임을 실행하여 총 t개의 부분열쇠들을 생성한다. 매단에서 리용되는 두개의 부분열쇠들은 어느 단에도 속하지 않는 더하기연산에 리용된다. 따라서 $t=2r+2$로 된다. 매 부분열쇠들은 길이가 한 단어(wbit)이다.

그림 4-11에 부분열쇠들을 생성하는 방법을 보여 주었다. 부분열쇠들은 S[0], S[1], ..., S[$t-1$]로 표식된 t개의 단어배렬로 분류된다. 입력으로서 파라메터 r, w를 리용하여 이 배렬을 어떤 고정된 우연비트패턴으로 초기화한다. 그다음 열쇠 K[0…$b-1$]은 C단어배렬 L[0…$c-1$]로 변환된다. 작은 말단기에서는 배렬 L을 0으로 초기화하고 문자렬 K를 직접 L에 의해 지적된 기억기위치에 복사한다. 만일 b가 w의 옹근수배가 아니면 L의 오른쪽 끝자리들에는 0을 채운다. 마지막으로 L의 내용들을 S의 초기화된 값에 적응시키는 혼합연산을 진행하여 배렬 S에 대한 마지막값을 생성한다.

이 연산을 보다 세부적으로 고찰하자.

초기화연산에서는 다음과 같이 정의되는 두개의 단어길이상수들을 리용한다.

$$P_w=\mathrm{Odd}[(e-2)2^w]$$
$$Q_w=\mathrm{Odd}[(\phi-1)2^w]$$

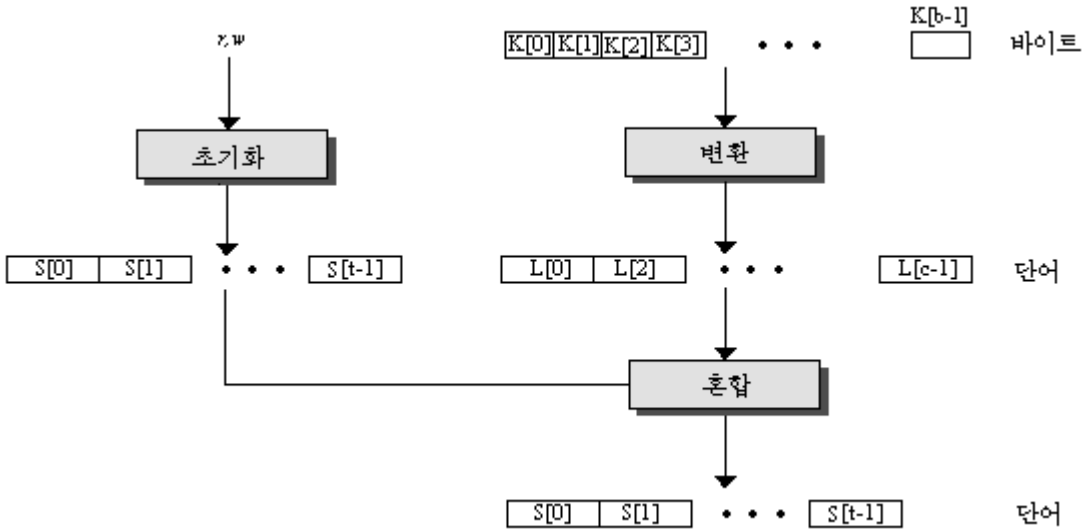



그림 4-11. RC5열쇠전개

여기서

$$e=2.718281828459\cdots(\text{자연로그의 밑수})$$
$$\phi=1.618033988749\cdots(\text{황금비}^{3})=(\frac{1+\sqrt{5}}{2})$$

이다(황금비는 수학에서 자주 쓰이는 수이다. 또한 π만큼 물리학분야에서 중요한 역할을 하는 수는 없는데 고대그리스의 수학자들은 우와 같은 수를 신비한 특징을 가진것으로 보고 신비한 비라고 불렀다).

이 Odd[x]는 x에 가장 가까운 홀수(x가 짝수이면 비록 여기서는 생기지 않지만 x우로의 둥그리기로 된다.)이다. 실례로 Odd[e]=3이고 Odd[ϕ]=1 이다.

w의 가능한 값들을 리용한 상수(16진수)들은 다음과 같다.

w	16	32	64
P_w Q_w	B7E1 9E37	B7E15163 9E3779B9	B7E151628AED2A6B 9E3779B97F4A7C15

이 두개의 상수들을 리용하여 배렬 S는 다음과 같이 초기화된다.

$S[0]=P_w$;
For i=1 to t−1 do
  $S[i]=S[i-1]+Q_w$;

여기서 더하기는 mod 2^w에 관하여 진행된다. 그때 초기화된 배렬 S는 부분열쇠들의 마지막배렬 S를 생성하기 위하여 배렬 L과 혼합된다. 이 과정은 다음과 같다.

$i = j = X = Y = 0$
do $3\times\max(t, c)$ times:
$S[i] = (S[i] + X + Y) <<< 3;\ X = S[i];\ i = (i+1) \bmod (t);$
$L[j] = (L[j] + X + Y) <<< (X+Y);\ Y = L[j];\ j = (j+1) \bmod (c);$

리베스트는 문헌[RIVE94]에서 열쇠확장함수는 한방향성을 가진다는것을 밝혔다. 즉 S로부터 K를 얻는것은 그리 쉽지 않다.

암호화

RC5는 3개의 기초연산(과 그의 거꿀)들을 리용한다.

- **더하기:** +로 표시되는 단어들의 더하기는 mod 2^w에 관하여 진행된다. 거꿀연산은 −로 표시되고 역시 mod 2^w에 관하여 진행된다.
- **비트별 배타적-OR :** 이 연산은 ⊕로 표시된다.
- **왼쪽순환회전:** 단어 x의 y에 의한 왼쪽순환밀기는 $x <<< y$로 표시된다. 거꿀은 단어 x의 y에 의한 오른쪽순환밀기로서 $x >>> y$로 표시된다.

그림 4-12의 ㄱ은 암호화연산을 보여 준다. 그림을 보고 알수 있는것처럼 이것은 고전페이스텔(Feistal)구조가 아니다. 평문은 처음 두개의 w-bit등록기 A, B로 나뉘여 진다. 여기서 i번째 단이 수행된후 자료를 왼쪽과 오른쪽으로 각각 절반씩 나누는데 변수 LE_i와 RE_i에 들어 간다. 알고리듬은 다음과 같은 가상부호에 의하여 정의된다.

$LE_0 = A + S[0];$
$RE_0 = B + S[1];$
for $i = 1$ to r do
$LE_i = ((LE_{i-1} \oplus RE_{i-1}) <<< RE_{i-1}) + S[2\times i];$
$RE_i = ((RE_{i-1} \oplus LE_i) <<< LE_i) + S[2\times i + 1];$

결과의 암호문은 두개의 변수 LE_r와 RE_r들에 포함된다. r개의 단들은 두개 단어의 자료를 리용하는 대입 및 치환열쇠에 의존하는 대입으로 이루어 진다. 연산은 아주 간단하며 5개 행의 부호로 정의된다. 또한 자료의 개개 절반부분들은 매단에서 갱신된다. 따라서 RC5의 한개 단은 DES의 두개 단과 부분적으로 류사하다.

복호화

그림 4-12의 ㄴ에서 보여 주는것처럼 복호화는 암호화알고리듬으로부터 쉽게 얻어진다. 이 경우 2$w$bit의 암호문은 처음 두개의 한단어변수들인 LD_r와 RD_r에 갈라 들어간다.

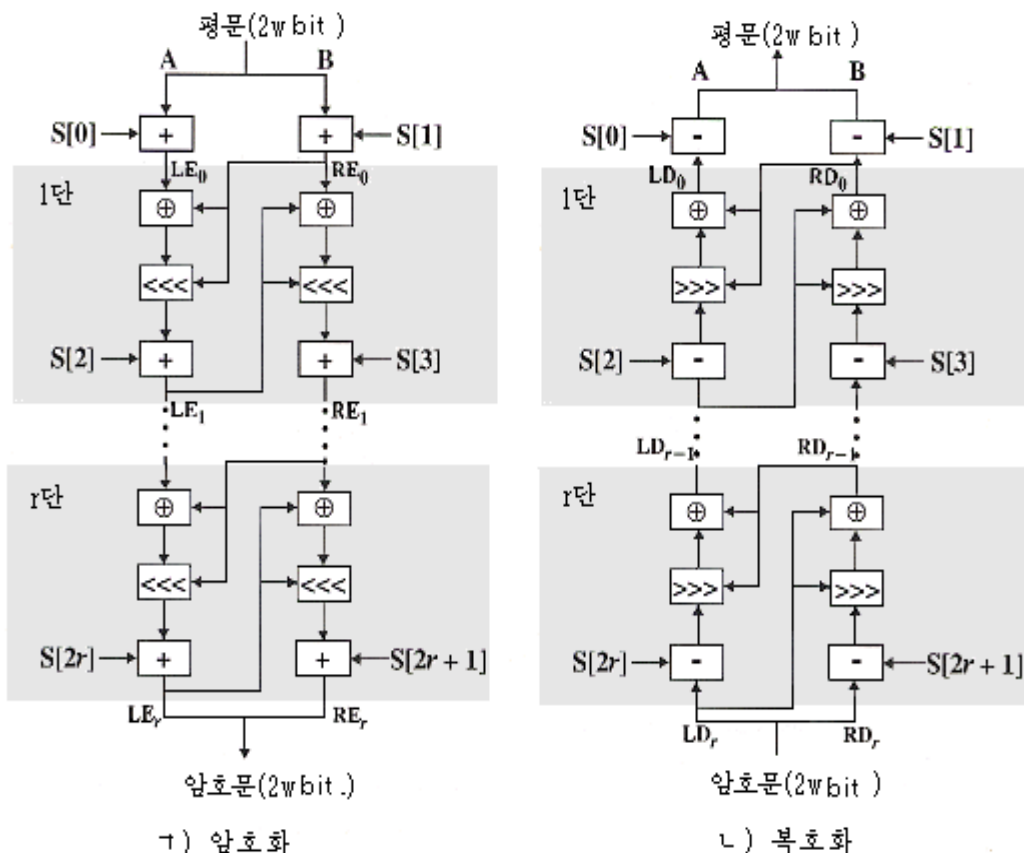



그림 4-12. RC5의 암호화 및 복호화

변수 LD_r와 RD_r는 i번째 단이 시작하기전의 자료의 왼쪽과 오른쪽 절반부분을 나타낸다. 여기서 단들은 r번째 단으로부터 첫번째 단으로 내려 가며 번호를 달았다.

for i = r down to 1 do

$RD_{i-1} = ((RD_i - S[2\times i+1] >>> LD_i) \oplus LD_i)$;

$LD_{i-1} = ((LD_i - S[2\times i] >>> RD_{i-1}) \oplus RD_{i-1})$;

$B = RD_0 - S[1]$;

$A = LD_0 - S[0]$;

RC5의 가장 중요한 두가지 측면은 알고리듬의 단순성과 자료의존회전의 리용이다. 회전은 알고리듬의 비선형부분이다. 리베스트는 회전들이 비선형적으로 자료값들에 의존하므로 선형 및 차분암호분석을 어렵게 한다고 보았다. 많은 연구자들이 이 주장을 확인하였다(문헌[YIN97]).

RC5 방식

필요에 따라 RC5의 효과성을 높이기 위하여 RFC 2040[BALD96]은 4개의 서로 다른 조작방식들을 정의하였다.

- **RC5블로크암호:** 이것은 고정된 크기의 입력블로크(2*w*bit들)를 가지고 열쇠에 의존하는 변환을 리용하여 같은 길이의 암호문블로크를 생성하는 암호화알고리듬이다. 이것은 흔히 전자부호책(Electronic Code Book: ECB)방식이라고도 한다.
- **RC5-CBC:** 이것은 RC5에 대한 암호블로크련쇄방식이다. CBC(암호블로크련쇄방식: cipher block chaining mode)는 3장에서 론의하였다(그림 3-12를 보시오). CBC는 RC5블로크크기의 배수길이(2*w* bit의 배수)의 통보들을 처리한다. 3장에서 보여 준것처럼 CBC는 평문에 대하여 반복블로크들이 서로 다른 암호블로크들을 생성하기때문에 ECB보다 강하다.
- **RC5-CBC-Pad:** 이것은 임의의 길이의 평문을 처리하는 CBC형식의 알고리듬이다. 암호문은 평문보다 기껏해서 하나의 RC5블로크크기만큼 길어 진다.
- **RC5-CTS:** 이것은 CBC암호형식으로 된 암호문강화방식인데 역시 CBC형의 알고리듬이다. 이 방식은 임의의 길이의 평문을 처리하여 같은 길이의 암호문을 생성한다.

우의 방식들에서 마지막 두개의 방식을 좀 더 구체적으로 설명하면 다음과 같다.

CBC방식으로써 통보문을 암호문으로 바꿀 때 어떤 경우에는 통보문의 길이가 블로크길이의 배수가 아닌 경우도 있다. 이것을 처리하기 위한 가장 단순한 방법은 메꾸기(padding)를 리용하는것이다. RC5에서는 통보문이 옹근수배바이트이라고 가정하였다. 통보문의 끝에 1부터 *bb*개까지의 바이트메꾸기가 추가된다. 여기서 *bb*는 RC5에서 바이트별로 측정한 블로크의 크기이다($bb=2w/8$). 메꾸기바이트들은 모두 같으며 메꾸기바이트수를 나타내는 바이트로 설정된다. 실례로 8개 바이트의 메꾸기가 있다면 매 바이트는 비트패턴 00001000을 가진다.

메꾸기는 언제나 쓰는것이 아니다. 실례로 처음에 평문이 들어 있던 기억완충기에 암호화된 자료를 보관하려고 하는 경우도 있다. 이 경우 암호문은 평문과 같은 길이여야 한다. RC5-CTS방식은 이러한 기능을 제공한다(그림 4-13). 평문의 마지막블로크의 길이가 *L*byte라고 하자. 여기서 $L<2w/8$이다. 암호화과정은 다음과 같다(RFC.2040에서 서술한것은 부정확하지만 여기서 쓴것은 정확하다).

1. CBC기술을 리용하여 첫 $(N-2)$개의 블로크들을 암호화한다.
2. Y_{N-1}을 생성하기 위하여 이전 암호문블로크 C_{N-2}과 P_{N-1}를 배타적론리합한다.
3. Y_{N-1}를 암호화하여 E_{N-1}를 얻는다.
4. C_N을 생성하기 위하여 E_{N-1}의 첫 *L*개 바이트를 선택한다.
5. P_N의 끝에 령으로써 메꾸기하고 E_{N-1}과 배타적론리합하여 Y_N을 얻는다.
6. Y_N을 암호화하여 C_{N-1}를 창조한다.

암호문의 마지막 두개의 블로크들은 C_{N-1}와 C_N이다.

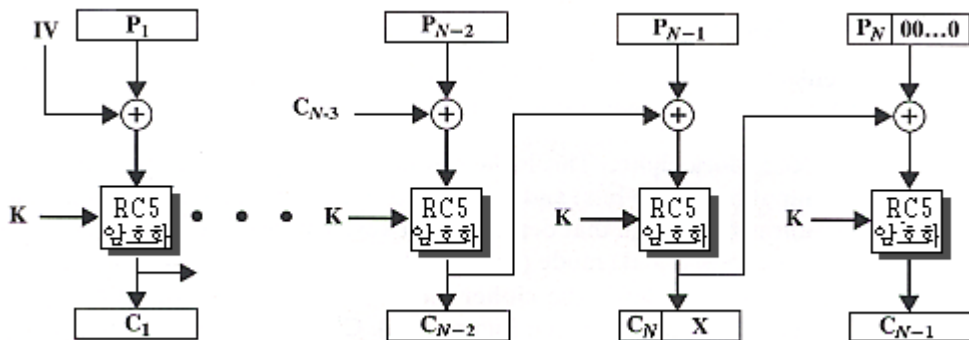



그림 4-13. RC5암호문강화방식

4.5 CAST-128

CAST는 카르리슬 아담스(Carlisle Adams)와 스태포드 타바레스(stafford Tavares)가 개발한 대칭블로크암호화알고리듬의 설계방식이다(문헌[ADAM97a]). 이 절에서는 CATS-128이라고 부르는 RFC2144에서 정의된(문헌[ADAM97b]) 알고리듬을 설명한다. CAST는 8bit씩 증가하는 40bit로부터 128bit까지의 가변열쇠를 리용한다.

CAST는 16단으로 되여 있으며 64-bit블로크의 평문을 처리하여 64-bit블로크의 암호문을 생성하는데 고전페이스텔(Feistel)구조로 되여 있다. 그림 3-5와의 차이점은 CAST가 매단에서 두개의 부분열쇠들 즉 32-bit Km_i와 5-bit Kr_i를 리용한다는것과 함수 F가 매단에 의존한다는 점이다.

CAST는 오랜 연구과정에 개발되였으므로 암호학적으로 확정된 연구결과이다. 이것은 PGP를 비롯하여 많은 제품들에서 리용되기 시작하였다.

여기서는 암호화알고리듬의 세부를 먼저 설명하고 그다음 부분열쇠생성과정을 설명한다.

CAST-128 암호화

CAST-128은 4개의 원시연산들을 리용한다.

- **더하기와 덜기:** 단어들의 더하기는 mod 2^{32}로 진행되며 +로 표기한다. 거꿀연산도 mod 2^{32}에서 진행되며 −로 표기한다.
- **비트별 배타적론리합:** 이 연산은 ⊕로 표기한다.
- **왼쪽순환밀기:** y에 의한 단어 x의 왼쪽순환밀기는 $x<<<y$로 표기한다.

CAST-128암호알고리듬은 다음과 같은 가상부호로 정의된다. 평문을 두개의 32bit 크기인 L_0과 R_0으로 나눈다. i번째 단이 완전히 수행된후의 자료는 왼쪽과 오른쪽 절반씩 갈라 변수 L_i와 R_i에 보관한다. 암호문은 16번째 단의 출력을 엇바꾸어 구성한다. 즉 R_{16}과 L_{16}의 련결이다.

$L_0||R_0$=Plaintext

For $i=1$ to 16 do

$L_i = R_{i-1}$;

$R_i = L_{i-1} \oplus F_i[R_{i-1}, km_i, Kr_i]$;

Ciphertext=$R_{16}||L_{16}$

복호화는 암호화와 같으며 다만 열쇠들을 거꿀순서로 리용한다.

그림 4-14에 한개 단의 세부를 보여 준다. F함수는 4개의 8×32 S-통들과 왼쪽순환밀기함수 그리고 $f1_i$, $f2_i$, $f3_i$, $f4_i$와 같이 단의 번호에 따라 변하는 4개 함수들을 포함한다. 왼쪽순환밀기함수후의 32-bit중간값을 참조하기 위하여 I를 리용하는데 I는 Ia, Ib, Ic, Id로 표시된다. 여기서 Ia가 제일 웃자리이고 Id가 제일 아래자리이다. 이와 같이 CAST-128의 매단 함수구조가 Blowfish보다 간단하다.

F에 대한 정의는 다음과 같다.

1,4,7,10,13,16단	$I=((Km_i + R_{i-1}) <<< Kr_i)$ $F=((S1[Ia] \oplus S2[Ib]) - S3[Ic]) + S4[Id]$
2,5,8,11,14단	$I=((Km_i \oplus R_{i-1}) <<< Kr_i)$ $F=((S1[Ia] - S2[Ib]) + S3[Ic]) \oplus S4[Id]$
3,6,9,12,15단	$I=((Km_i - R_{i-1}) <<< Kr_i)$ $F=((S1[Ia] + S2[Ib]) \oplus S3[Ic]) - S4[Id]$

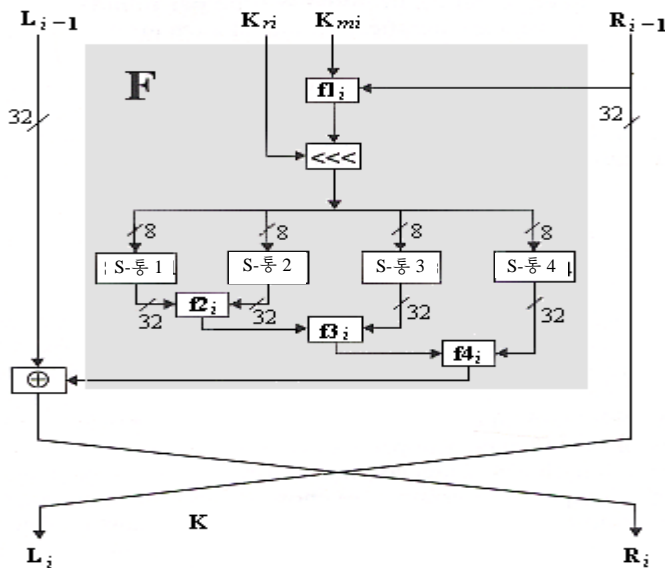



그림 4-14. 하나의 CAST-128단의 세부

대입통

CAST-128은 8개의 8×32 S-통들을 리용한다. 그중 4개 즉 S-통1로부터 S-통4까지는 암호화와 복호화처리에서 리용된다. 나머지 4개 즉 S-통5부터 S-통8까지는 부분열쇠생성에 리용된다. 매개 S-통은 32개의 렬들로 된 배렬인데 256개의 행들을 가진다. 8-bit입력은 행렬의 행을 선택하는데 그 행의 32-bit값이 출력이다.

모든 S-통들은 고정된 값을 가지고 있다. S-통설계자들은 고차의 비선형성을 가지도록 하기 위하여 3장에서와 부록 3A에서 서술한 굴곡함수(bent function)를 리용하였다.

부분열쇠생성

부분열쇠생성은 복잡하다. 128-bit열쇠를 바이트별로 표식하기 위하여 다음과 같이 한다.

x0x1x2x3x4x5x6x7x8x9xAxBxCxDxExF

여기서 x0이 제일 웃자리바이트이고 xF는 제일 아래자리바이트이다.

다음과 같은 기호들을 정의한다.

$Km_1, \cdots, Km_{16}$	16개의 32-bit회전부분열쇠들(매단당 한개)
$Kr_1, \cdots, Kr_{16}$	16개의 32-bit회전부분열쇠(매단당 한개)들로써 매개의 제일 아래자리 5개 비트들만을 리용한다.
z0...zF	중간(림시)바이트들
K1,..., K32	중간(림시)32-bit단어들

그림 4-15에 S-통5부터 S-통8까지를 리용하여 값 K_1부터 K_{32}까지가 열쇠로부터 어떻게 계산되는가를 보여 주었다.

이때 부분열쇠들은 다음과 같이 정의된다.

$$\begin{aligned}&\text{for } i=1 \text{ to } 16 \text{ do}\\&\quad K_{mi}=K_i;\\&\quad K_{ri}=K_{16+i};\end{aligned}$$

론의

CAST-128에는 몇가지 주목할만한 특징들이 있다. 첫째로, CAST는 고정된 S-통들을 리용한다. CAST설계자들은 고정된 S-통을 좋은 비선형특성을 가지고 열쇠에 의존하도록 한다면 S-통들을 우연화할수 있다고 보았다. CAST를 리용한 실례를 [ADAM97a]에서 서술하였으며 보다 자세히는 [MIST96]에서 론의하였다. 여기서는 그에 대하여 구체적으로 보지 않고 다음과 같은 수속만을 소개한다. 길이가 256인 32개의 서로 다른 2진굴곡백토르를 선택한다. 매 행벡토르는 어떤 주어 진 입력에 대하여 하나의 출력값을 나타낸다. 벡토르들은 그 합이 비선형적이며(mod 2로써) 좋은 사태특성을 가지도록 선

택되였다. 이 처리에는 한번에 하나의 렬선택과 선택된 렬들의 모임에 대한 검사 그리고 또 다른 선택을 하거나 또는 다음 렬로 이행하는것들이 포함된다.

CAST-128에서 리용한 부분열쇠생성처리는 대칭블로크암호들에서 리용한것과 다르다.

```
z0z1z2z3 = x0x1x2x3 ⊕ S5[xD] ⊕ S6[xF] ⊕ S7[xC] ⊕ S8[xE] ⊕ S7[x8]
z4z5z6z7 = x8x9xAxB ⊕ S5[z0] ⊕ S6[z2] ⊕ S7[z1] ⊕ S8[z3] ⊕ S8[xA]
z8z9zAzB = xCxDxExF ⊕ S5[z7] ⊕ S6[z6] ⊕ S7[z5] ⊕ S8[z4] ⊕ S5[x9]
zCzDzEzF = x4x5x6x7 ⊕ S5[zA] ⊕ S6[z9] ⊕ S7[zB] ⊕ S8[z8] ⊕ S6[xB]
K1  = S5[z8] ⊕ S6[z9] ⊕ S7[z7] ⊕ S8[z6] ⊕ S5[z2]
K2  = S5[zA] ⊕ S6[zB] ⊕ S7[z5] ⊕ S8[z4] ⊕ S6[z6]
K3  = S5[zC] ⊕ S6[zD] ⊕ S7[z3] ⊕ S8[z2] ⊕ S7[z9]
K4  = S5[zE] ⊕ S6[zF] ⊕ S7[z1] ⊕ S8[z0] ⊕ S8[zC]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
K5  = S5[x3] ⊕ S6[x2] ⊕ S7[xC] ⊕ S8[xD] ⊕ S5[x8]
K6  = S5[x1] ⊕ S6[x0] ⊕ S7[xE] ⊕ S8[xF] ⊕ S6[xD]
K7  = S5[x7] ⊕ S6[x6] ⊕ S7[x8] ⊕ S8[x9] ⊕ S7[x3]
K8  = S5[x5] ⊕ S6[x4] ⊕ S7[xA] ⊕ S8[xB] ⊕ S8[x7]
z0z1z2z3 = x0x1x2x3 ⊕ S5[xD] ⊕ S6[xF] ⊕ S7[xC] ⊕ S8[xE] ⊕ S7[x8]
z4z5z6z7 = x8x9xAxB ⊕ S5[z0] ⊕ S6[z2] ⊕ S7[z1] ⊕ S8[z3] ⊕ S8[xA]
z8z9zAzB = xCxDxExF ⊕ S5[z7] ⊕ S6[z6] ⊕ S7[z5] ⊕ S8[z4] ⊕ S5[x9]
zCzDzEzF = x4x5x6x7 ⊕ S5[zA] ⊕ S6[z9] ⊕ S7[zB] ⊕ S8[z8] ⊕ S6[xB]
K9  = S5[z3] ⊕ S6[z2] ⊕ S7[zC] ⊕ S8[zD] ⊕ S5[z9]
K10 = S5[z1] ⊕ S6[z0] ⊕ S7[zE] ⊕ S8[zF] ⊕ S6[zC]
K11 = S5[z7] ⊕ S6[z6] ⊕ S7[z8] ⊕ S8[z9] ⊕ S7[z2]
K12 = S5[z5] ⊕ S6[z4] ⊕ S7[zA] ⊕ S8[zB] ⊕ S8[z6]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
K13 = S5[x8] ⊕ S6[x9] ⊕ S7[x7] ⊕ S8[x6] ⊕ S5[x3]
K14 = S5[xA] ⊕ S6[xB] ⊕ S7[x5] ⊕ S8[x4] ⊕ S6[x7]
K15 = S5[xC] ⊕ S6[xD] ⊕ S7[x3] ⊕ S8[x2] ⊕ S7[x8]
K16 = S5[xE] ⊕ S6[xF] ⊕ S7[x1] ⊕ S8[x0] ⊕ S8[xD]

z0z1z2z3 = x0x1x2x3 ⊕ S5[xD] ⊕ S6[xF] ⊕ S7[xC] ⊕ S8[xE] ⊕ S7[x8]
z4z5z6z7 = x8x9xAxB ⊕ S5[z0] ⊕ S6[z2] ⊕ S7[z1] ⊕ S8[z3] ⊕ S8[xA]
z8z9zAzB = xCxDxExF ⊕ S5[z7] ⊕ S6[z6] ⊕ S7[z5] ⊕ S8[z4] ⊕ S5[x9]
zCzDzEzF = x4x5x6x7 ⊕ S5[zA] ⊕ S6[z9] ⊕ S7[zB] ⊕ S8[z8] ⊕ S6[xB]
K17 = S5[z8] ⊕ S6[z9] ⊕ S7[z7] ⊕ S8[z6] ⊕ S5[z2]
K18 = S5[zA] ⊕ S6[zB] ⊕ S7[z5] ⊕ S8[z4] ⊕ S6[z6]
K19 = S5[zC] ⊕ S6[zD] ⊕ S7[z3] ⊕ S8[z2] ⊕ S7[z9]
K20 = S5[zE] ⊕ S6[zF] ⊕ S7[z1] ⊕ S8[z0] ⊕ S8[zC]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
K21 = S5[x3] ⊕ S6[x2] ⊕ S7[xC] ⊕ S8[xD] ⊕ S5[x8]
K22 = S5[x1] ⊕ S6[x0] ⊕ S7[xE] ⊕ S8[xF] ⊕ S6[xD]
K23 = S5[x7] ⊕ S6[x6] ⊕ S7[x8] ⊕ S8[x9] ⊕ S7[x3]
K24 = S5[x5] ⊕ S6[x4] ⊕ S7[xA] ⊕ S8[xB] ⊕ S8[x7]
z0z1z2z3 = x0x1x2x3 ⊕ S5[xD] ⊕ S6[xF] ⊕ S7[xC] ⊕ S8[xE] ⊕ S7[x8]
z4z5z6z7 = x8x9xAxB ⊕ S5[z0] ⊕ S6[z2] ⊕ S7[z1] ⊕ S8[z3] ⊕ S8[xA]
z8z9zAzB = xCxDxExF ⊕ S5[z7] ⊕ S6[z6] ⊕ S7[z5] ⊕ S8[z4] ⊕ S5[x9]
zCzDzEzF = x4x5x6x7 ⊕ S5[zA] ⊕ S6[z9] ⊕ S7[zB] ⊕ S8[z8] ⊕ S6[xB]
K25 = S5[z3] ⊕ S6[z2] ⊕ S7[zC] ⊕ S8[zD] ⊕ S5[z9]
K26 = S5[z1] ⊕ S6[z0] ⊕ S7[zE] ⊕ S8[zF] ⊕ S6[zC]
K27 = S5[z7] ⊕ S6[z6] ⊕ S7[z8] ⊕ S8[z9] ⊕ S7[z2]
K28 = S5[z5] ⊕ S6[z4] ⊕ S7[zA] ⊕ S8[zB] ⊕ S8[z6]
x0x1x2x3 = z8z9zAzB ⊕ S5[z5] ⊕ S6[z7] ⊕ S7[z4] ⊕ S8[z6] ⊕ S7[z0]
x4x5x6x7 = z0z1z2z3 ⊕ S5[x0] ⊕ S6[x2] ⊕ S7[x1] ⊕ S8[x3] ⊕ S8[z2]
x8x9xAxB = z4z5z6z7 ⊕ S5[x7] ⊕ S6[x6] ⊕ S7[x5] ⊕ S8[x4] ⊕ S5[z1]
xCxDxExF = zCzDzEzF ⊕ S5[xA] ⊕ S6[x9] ⊕ S7[xB] ⊕ S8[x8] ⊕ S6[z3]
K29 = S5[x8] ⊕ S6[x9] ⊕ S7[x7] ⊕ S8[x6] ⊕ S5[x3]
K30 = S5[xA] ⊕ S6[xB] ⊕ S7[x5] ⊕ S8[x4] ⊕ S6[x7]
K31 = S5[xC] ⊕ S6[xD] ⊕ S7[x3] ⊕ S8[x2] ⊕ S7[x8]
K32 = S5[xE] ⊕ S6[xF] ⊕ S7[x1] ⊕ S8[x0] ⊕ S8[xD]
```

그림 4-15. CAST-128 부분열쇠생성

CAST설계자들은 가능한껏 암호분석을 막는데 중심을 두고 이 강도보장을 위하여 비선형성이 강한 S-통들을 리용하였다고 보아 진다. 앞에서도 이러한 목적을 지향한 방법들을 보았다. 실례로 Blowfish는 부분열쇠생성에 자기자체의 암호화알고리듬을 리용한다. RC5는 가변길이의 회전들과 mod 2로의 더하기를 포함하는 연산들의 복합모임으로서 얻어 지는 준우연초기화렬을 리용한다.

이 방법들중 어느것이 우월한가는 가늠하기 어렵다. 그러나 그것들은 어느것이나 DES에서 리용한 간단한 대입-치환형식보다는 암호강도를 더 크게 한다고 볼수 있다.

함수 F는 좋은 혼란, 확산, 련쇄특성을 가지도록 설계되였다. F는 S-통, mod 2더하기 및 덜기, 배타적론리연산, 열쇠에 의존하는 밀기를 리용한다. F함수의 강도는 초시기 S-통의 강도에 의존하였지만 앞으로는 이 산수적인 방법들의 리용에 의존할것이다. 론리연산자와 순환연산자들은 강도를 더 높인다. 단의 번호에 따르는 F의 의존성은 증명되지는 않았지만 강도를 더 높게 할것이다.

4.6 RC2

RC2는 론 리베스트(Ron Rivest)[RIVE97]에 의하여 개발된 대칭블로크암호이다. RC2은 64bit길이의 암호문과 평문블로크를 리용하여 8bit부터 1024bit까지에서 변하는 가변적인 열쇠크기를 가진다. 알고리듬은 그것이 설계된 시대를 반영하여 16-bit극소형처리기들에서 쉽게 실현할수 있도록 설계되였다. RC2는 40, 64, 128-bit열쇠크기를 가지며 S/MIME에서 리용된다.

열쇠확장

RC5는 바이트의 부분열쇠들을 생성하며 비밀열쇠들에 대하여 연산들을 진행한다. 부분열쇠들은 L[0], L[1], …, L[127]byte배렬에 보관된다. 일부 연산들에서 16-bit단어배렬 K[0], K[1] ,…, K[63]과 같은 부분열쇠들을 참조하는것이 편리하다.

T byte의 열쇠가 제공된다고 하자. 여기서 $1 \le T \le 128$이다. 열쇠생성은 열쇠의 T개 바이트들을 L[0], …, L[T]byte들에 넣는것으로부터 시작한다. 그다음 L배렬은 임의의 준우연바이트들의 보조배렬 P[0], P[1], …, P[255]를 리용하여 계산되는데 이 배렬은 π값에 따른다. 계산과정은 다음과 같다.

```
for i = T to 127 do                      /*set L[T]through L[127]*/
        L[i]=P[L[i-1]+L[i-T]];
L[128-T]=P[L[128-T]
for i =127-T down to 0 do                /*set L[0] through L[127-T]*/
        L[i]=P[L[i+1]⊕L[i]+T]];
```

일반적으로 첫단은 이전 부분열쇠바이트와 뒤에 있는 부분열쇠바이트의 T개의 위치들에서의 합에 대한 첫 T개 바이트다음의 매개 확장된 부분열쇠바이트를 설정한다. 두번째 단은 다음부분열쇠바이트와 앞에 있는 부분열쇠바이트의 T개의 위치들에서 XOR에 대하여 마지막 T개 바이트들을 제외한 매개 부분열쇠바이트를 설정한다.

암호화

RC2는 다음과 같은 원시연산들을 리용한다:

- **더하기:** 단어단위의 더하기로서 +로 표시되며 mod 2^{32}의 더하기를 진행한다. 거꿀연산은 −로 표시되며 역시 mod 2^{32}의 덜기이다.
- **비트별 배타적론리합:** 이 연산은 ⊕로 표시된다.
- **비트별 보수:** 이 연산은 ~로 표시된다.
- **비트별 AND:** 이 연산은 &로 표시된다.
- **왼쪽순환밀기:** y bit들에 의한 단어 x의 왼쪽순환밀기는 $x <<< y$로 표시한다.

이 책에서 서술한 다른 대칭블로크암호들과 달리 RC2는 고전페이스텔(Feistel)구조를 리용하지 않는다. 이것으로 해서 다른 알고리듬들과 비교하기 어렵다.

암호알고리듬은 16-bit단어 R[0],R[1],R[2],R[3]들에 보관된 64-bit입력를 가지며 그 결과들을 R[0]부터 R[3]에 다시 넣는다. 알고리듬은 혼합과 가름이라는 두 가지 형태로 된 총 18개 단으로 구성되여 있다. 혼합단(mixing round)은 다음과 같다.

R[0]=R[0]+K[j]+(R[3]&R[2])+((~R[3]&R[1]);
R[0]=R[0]<<<1;
$j=j+1$;
R[1]=R[1]+K[j]+(R[0]&R[3])+((~R[0]&R[2]);
R[1]=R[1]<<<2;
$j=j+1$;
R[2]=R[2]+K[j]+(R[1]&R[0])+((~R[1]&R[3]);
R[2]=R[2]<<<3;
$j=j+2$;
R[3]=R[3]+K[j]+(R[2]&R[1])+((~R[2]&R[0]);
R[3]=R[3]<<<5;
$j=j+3$;

이 식에서 K[j]는 아직 리용되지 않은 첫번째 부분열쇠단어이다.

이 연산을 다음과 같이 설명할수 있다. 매 단어 R[i]에 대하여 다음부분열쇠단어 K[j]는 R[i]에 더해 진다. 그다음 R에 대한 첨수가 mod 3인 R[$i-1$]은 R[i]에 추가되는 합성단어를 창조하는데 리용된다. 합성단어는 R[$i-1$]이 1인 비트자리들에서 R[$i-2$]의 비트들과 R[$i-1$]이 0인 비트자리들에 놓인 R[$i-3$]의 비트들로 구성된다. 그다음 R[i]를 왼쪽순환밀기하며 j를 증가시킨다.

가르기단(mashing round)은 다음과 같다.

R[0]=R[0]+K[R[3] & 63];
R[1]=R[1]+K[R[0] & 63];
R[2]=R[2]+K[R[1] & 63];
R[3]=R[3]+K[R[2] & 63];

말하자면 매 R[i]에 대하여 부분열쇠단어는 R[i]에 추가된다. 부분열쇠배렬은 R[i-1]의 낮은 자리 6bit들에 의하여 첨수화된다. RC2는 이때 다음과 같이 정의된다. 매 단계후의 j값은 괄호안에 표시해 주었다.

1. j를 0으로 초기화한다.
2. 5개의 혼합단들을 수행한다(j=20).
3. 하나의 가름단을 수행한다.
4. 6개의 혼합단을 수행한다(j=44).
5. 하나의 가름단을 수행한다.
6. 5개의 혼합단들을 수행한다(j=64).

매 혼합단에는 4개의 부분열쇠단어들을 리용한다. 16개의 혼합단들이 있으므로 모든 부분열쇠들은 한번 리용된다. 그외 부분열쇠들은 가름단에 대하여 자료의존형식으로 선택된다.

복호화는 암호화의 거꿀과정으로 진행되므로 매단들과 열쇠들은 거꿀순서로 리용된다.

4.7 개선된 대칭블로크암호의 특성

사실 모든 대칭블로크암호들은 많은 면에서 DES 및 기초페이스텔(Feistel)블로크암호구조와 류사하다. 그것은 DES설계에 참가한 IBM과 NSA가 합의했기때문이다. 그러나 암호분석과 고속쏘프트웨어암호에 대한 요구에 따라 커다란 전진이 이룩되였다. 이 절에서는 가장 중요한 현대대칭암호알고리듬들에서의 그러한 성과들에 대하여 설명하였다. 이 부분에서는 그 알고리듬들에서 찾아 볼수 있는 DES에는 없는 몇가지 열쇠특징들에 대하여 강조한다.

- **가변열쇠길이:** 만일 암호알고리듬이 암호분석을 어렵게 하도록 구성되였다면 그 강도는 열쇠길이에 의하여 결정된다. 열쇠길이가 길수록 열쇠탐색을 더 어렵게 한다. Blowfish, RC5, CAST-128, RC2는 열쇠길이가 가변이다.
- **혼합연산자:** 한개이상의 산수적 및/또는 론리연산자들이 분배법칙과 결합법칙들을 만족시키지 않을 때 암호분석은 복잡해 진다. 이 수법은 S-통의 대안으로서 비선형성을 제공한다. 3중DES를 제외하고 이 장의 모든 알고리듬들은 혼합연산자를 리용한다.
- **자료-의존회전:** S-통에 대한 다른 대안은 자료에 의존하는 순환들을 리용하는것이다. 충분한 수의 단들에 대하여 이것은 좋은 혼란과 확산을 제공한다. 더우기 순환들은 부분열쇠들이 아니라 단들을 거쳐 움직이는 자료블로크들에 의존한다. 이것은 부분열쇠들의 회복을 더욱 어렵게 한다. RC5는 자료-의존의 회전들을 리용한다.
- **열쇠-의존회전:** 회전은 자료보다 오히려 열쇠에 의존하는것을 리용한다.
- **열쇠-의존 S-통:** DES와 CAST-128에서와 같이 요구하는 암호학적특성들을 가지는 고정된 S-통들을 설계하기보다 S-통들의 내용들이 열쇠에 의존될수 있다. 서로 다른 열쇠들은 서로 다른 S-통들을 만든다. 이 수법은 특히 더 큰 S-통(실례로 8×32)에 대하여 고도의 비선형적인 결과들을 주며 암호분석을 매우 어렵게 한다. Blowfish는 열쇠의존형 S-통들을 리용한다.

- **긴열쇠알고리듬:** 이것은 Blowfish에서 리용되는 독특한 수법이다. 부분열쇠들의 생성은 한개의 암호화 또는 부호화보다 훨씬 더 오래 걸린다. 그 결과 전수탐색공격의 성과가 비상히 커진다.
- **가변 F:** 단으로부터 단으로의 이행에서 변하는 함수 F의 리용은 암호분석을 복잡하게 할수 있다. CAST-128은 가변 F를 리용한다.
- **가변의 평문/암호문블로크길이:** 블로크길이가 길면 길수록 더 큰 암호강도를 제공한다. 물론 단들의 수의 증가도 암호화/복호화시간을 증대시킨다. 단들의 수를 가변으로 하면 사용자가 보안과 실행속도사이의 절충을 만들수 있다. RC5는 단의 수가 가변인 단들을 제공한다.
- **매단에서 두개의 반자료들에 대한 조작:** 고전페이스텔(Feistel)암호에서는 자료의 한개의 절반부분만이 매단에서 교체된다. 만일 간단한 조작이 교체되지 않은 다른 절반부분에서 진행되면 보안은 실행시간에서 적게 증가할것이다. IDEA, Blowfish 및 RC5는 매단에서 자료의 두개의 절반부분들에 적용한다. 현재 연구의 초점은 이 장에서 설명한 내용들 즉 완전히 새로운 구조를 설계하는것보다 고전페이스텔(Feistel)암호와 DES에 대한 개선을 위한 방책들을 찾는데 두고 있다. 그것은 페이스텔(Feistel)구조가 많이 연구되고 있지만 고유한 약점들이 발로되지 않고 있기때문이다.

문 제

1. DES보다 더 강한 알고리듬을 리용하여 암호블로크련쇄(CBC)방식으로 블로크암호화를 실현하는 장치를 만들려고 한다. 3중DES는 훌륭한 후보로 되지만 그것은 ECB방식에서만 정의된다. 3중DES을 위한 CBC방식에 대하여 규격이 설정되지 않았지만 그림 4-16에 두 경쟁자들을 보여 주었는데 그들모두는 CBC의 정의에 따른다. 어느것을 선택하겠는가?
   ㄱ) 보안을 위하여?
   ㄴ) 실행(속도)을 위하여?
2. 세개의 DES소편들과 몇개의 XOR기능들만을 리용하여 그림 4-16에 있는 다른 방법에 대한 보안향상을 제안할수 있는가? 두개의 열쇠로 제한된다고 가정하시오.
3. 3중DES에 대한 머클 헬만(Merkle-Hellman)공격은 A=0(그림 4-16)의 값을 가정하는것으로부터 시작된다. 그러면 2^{56}개의 가능한 K_1의 매개 값들에 대하여 A=0을 발생하는 평문 P가 결정된다. 그 알고리듬의 나머지부분을 설명하시오.
4. IDEA의 실행에서 제일 힘든 부분은 mod $(2^{16}+1)$에 관한 곱하기이다. 다음의 등식은 효률적인 실행을 달성할수 있는 방법을 준다. a, b가 령 아닌 두개의 n bit의 옹근수들이라고 하면

$$ab \bmod (2^n+1) = \begin{cases} (ab \bmod 2^n) - (ab \operatorname{div} 2^n) & ab \bmod 2^n \geq ab \operatorname{div} 2^n \text{일 때} \\ (ab \bmod 2^n) - (ab \operatorname{div} 2^n) + 2^n + 1 & ab \bmod 2^n \geq ab \operatorname{div} 2^n \text{일 때} \end{cases}$$

$(ab \bmod 2^n)$은 ab의 제일 아래자리 n bit이고 $(ab \operatorname{div} 2^n)$은 ab를 n bit오른쪽으로 밀기한것이다. 이 문제의 목적은 우의 같기식이 참이라는것을 증명하는것이다.

ㄱ) $ab = q(2^n + 1) + r$인 부 아닌 옹근수 q와 r가 있다는것을 증명하시오.
ㄴ) q와 r의 웃경계와 아래경계는 무엇인가?
ㄷ) $q + r < 2^{n+1}$이라는것을 증명하시오.
ㄹ) q를 리용하여 $(ab \text{ div } 2^n)$에 대한 식을 유도하시오.
ㅁ) q와 r로 $(ab \bmod 2^n)$ 에 대한 식을 유도하시오.
ㅂ) ㄹ과 ㅁ의 결과들을 리용하여 r에 대한 식을 유도하시오.
ㅅ) r가 이 문제의 첫 등식의 오른쪽과 같다는것을 증명하시오.

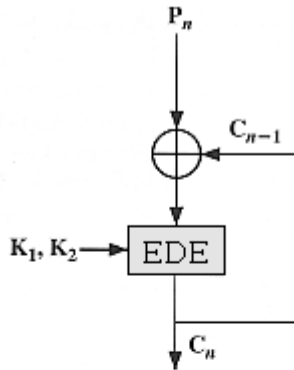
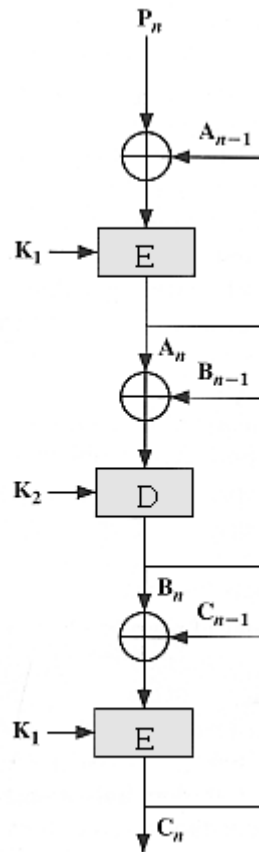



그림 4-16. CBC방식에서 3중DES의 리용

5. IDEA는 완전한 확산을 제공하는데 4개의 연산들을 리용한다(그림 4-3). 이것이 요구되는 최소연산수이라는것을 밝히시오. 이를 위하여 (Z_1, Z_2)의 매 선택에 대하여 $E(', ', Z_1, Z_2)$이 비가역이도록 다음과 같은 형태의 함수를 고찰한다.

$$(Y_1, Y_2) = E(X_1, X_2, Z_1, Z_2) \qquad 0 \le X_i, Y_i \le 2^m;\ 0 \le Z_i \le 2^k$$

만일 암호함수의 매개 출력변수들이 매 입력변수들에 의존하면 그 함수는 완전확산을 가진다고 말한다. 암호함수가 우의 형식과 같고 완전확산을 가지면 그 알고리듬이 적어도 4개의 연산들을 포함한다는것을 증명할수 있다.

ㄱ) 함수가 적어도 세개의 연산들을 포함해야 한다는것을 밝히시오.

ㄴ) E가 정확히 세개의 연산들을 가진다는것을 가정하고 이러한 함수가 비가역일수 없다는것을 밝히시오.

6. ㄱ) IDEA의 곱하기연산에 왜 간단히 mod 2^{16}대신에 mod $2^{16}+1$을 리용하는가?

   ㄴ) IDEA의 더하기에서는 왜 mod $2^{16}+1$대신에 mod 2^{16}을 리용하는가?

7. 라이(Lai)와 마쎄이(Massey)에 의하여 PES라고 표시된 암호알고리듬에 대한 원래의 제안이 IDEA와 다른 점들은 다음과 같다.

   1. 그림4-5의 우의 회색통의 4개의 함수들이 IDEA에서는 ⊙, ⊞, ⊞, ⊙의 순서로 있지만 PES에서는 ⊙, ⊙, ⊞, ⊞의 순서로 있다.
   2. IDEA에서는 매단을 거친 다음 두번째와 세번째 블로크들이 교체된다. PES에서는 매개 단을 거친 다음 첫번째와 두번째 블로크들이 세번째와 네번째 블로크들과 교체된다.

   두번째 변경은 차분분석([LAI91]을 보시오.)에 대한 저항을 증가시킨다는것을 알수 있다. 첫번째 변화는 암호강도에서 아무런 변화도 일으키지 못한다는것을 밝히시오.

8. Blowfish의 복호화가 Blowfish의 암호화와 거꿀이라는것을 밝히시오.
9. RC5의 복호화가 RC5의 암호화와 거꿀이라는것을 밝히시오.
10. RC5－CBC-Pad방식에서는 하나부터 *bb* byte까지의 메꾸기가 있다. 왜 령바이트들의 메꾸기가 허락되지 않는가? 즉 암호화할 통보문이 블로크크기의 옹근수배이면 왜 메꾸기를 하지 않는가?
11. 그림 4-17은 평문이 블로크크기의 옹근수배가 아닐 때 평문의 길이와 같은 암호문을 생성하는 RC5-CTS에 대한 다른 방법을 보여 준다.

    ㄱ) 알고리듬을 설명하시오.

    ㄴ) 왜 RC5-CTS가 그림 4-17에서 설명되는 수법에 비해 더 적합한가?
12. RC5-CTS방식에서 C_{n-1}과 C_n을 어떻게 복호하는가?
13. CAST-128의 복호화가 CAST-128의 암호화와 거꿀이라는것을 밝히시오.

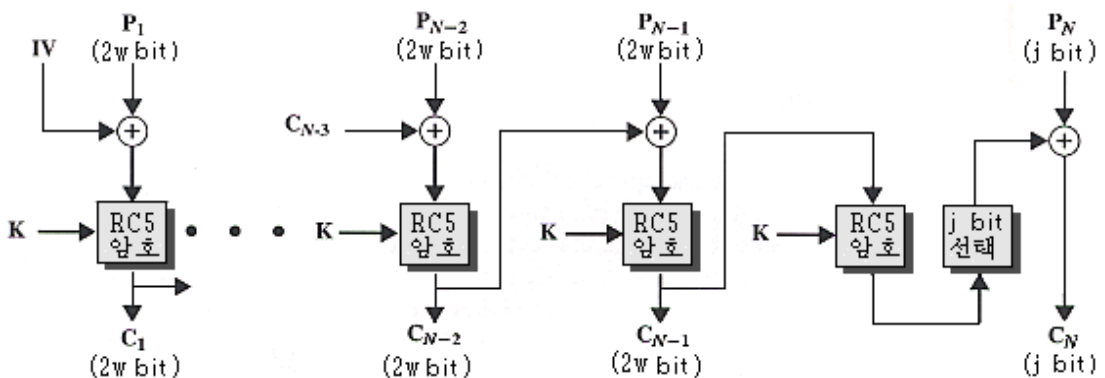



그림 4-17. RC5 CBC방식에서 마지막 짧은 블로크의 암호화방법

제5장. 전통암호에 의한 기밀성

력사적으로 암호학의 초점은 비밀을 보장하기 위하여 전통암호를 리용하는데 집중되여 왔다. 인증, 완정성, 수자서명, 공개열쇠암호의 리용과 같은 기밀성을 위한 방법들의 응용은 불과 수십년동안에 리론과 실천분야에 포괄되게 되였다.

이와 같은 보다 최근의 문제들을 취급하기전에 이 장에서는 먼저 기밀성을 보장하기 위한 전통암호의 리용에 대하여 고찰한다. 그것은 전통암호에 대한 리해가 공개열쇠암호의 개발에서 도움으로 되며 인증과 같은 암호학의 다른 응용들에 포함된 여러 개념들을 더 명확히 해주기때문이다.

암호학의 위치에 대한 론의로부터 시작하자. 여기서 기본선택은 접속으로 알려 진것과 말단 대 말단암호사이에 있다. 다음으로 전송정보분석공격을 막기 위한 암호의 리용을 취급한다. 그리고 열쇠배포의 복잡한 문제와 마지막으로 기밀성편의도구를 제공하는데서 중요한 도구의 기초로 되고 있는 원리(란수발생)를 론의한다.

5.1 암호기능의 설치

암호가 기밀성에 대한 공격을 막는데 리용되자면 우선 무엇을 암호화하고 암호기능을 어디에 두어야 하겠는가가 결정되여야 한다. 이 절에서는 보안공격의 가능한 위치와 암호설치의 두가지 기본수법 즉 접속(Link)과 말단 대 말단에 대하여 보게 된다.

기밀성에서 공격의 대상으로 될수 있는 위치

일반 봉사기관들에서의 사용자워크스테이션을 실례로 보자. 그림 5-1에 워크스테이션에서 리용될수 있는 통신수단들의 형태들과 공격 받을수 있는 위치를 주었다.

대부분의 기관들에서 워크스테이션을 국부망(LAN)에 접속시킨다. 일반적으로 사용자는 다리와 경로조종기로 접속된 한 건물에서 다른 LAN 혹은 자기의 LAN우의 워크스테이션, 가입자, 봉사기에 직접적으로 망통신할수 있다. 이 경우 기본문제는 다른 직원에 의한 도청이다. 일반적으로 LAN은 방송망이다. 한 국으로부터 다른 국으로의 전송은 모든 국들에서 LAN의 매개물을 통해 볼수 있다. 자료는 프레임형식으로 전송되며 매 프레임은 발신인(원천지)과 수신인(목적지)주소를 포함하고 있다. 도청자는 LAN에서의 통신을 감시할수 있으며 발신인과 수신인의 주소에 기초하여 주목하는 임의의 통신내용을 획득할수 있다.

더우기 도청자는 그 건물안의 직원만이 될수 있는것이 아니다. 만일 LAN이 LAN우의 통신봉사기들이나 가입자중의 하나를 통하여 전화접속(Dial-in)기능을 제공하면 침입자는 그 LAN에 접근할수 있다.

LAN으로부터 외부에로의 접근은 경로조종기, 전화접속모뎀들의 렬 또는 다른 형태의 봉사기형식으로 늘 가능하다. 통신봉사기로부터 배선실(wiring closet)까지 회선이 있게 된다. 배선실은 내부자료와 전화회선을 결합시키며 외부통신을 위한 결속점을 마련하는 집결점으로 된다.

배선실자체는 공격당하기 쉽다. 만일 침입자가 배선실에 침투하면 그는 자료통신에 어느 회선이 리용되는가를 결정하기 위해 매개 회선을 도청할수 있다. 하나 또는 여러개

회선을 중단시킨후 침입자는 저출력라지오송신기를 설치할수 있다. 가까운 곳에서 송신하는 신호들을 얻어 낼수 있다(근방의 건물, 주차한 차량).

배선실로부터 나가는 여러 경로들이 있을수 있다. 표준적구성은 지역전화회사의 가장 가까운 중심사무소에 대한 접근을 제공한다. 배선실에서 회선들은 케블로 통합되며 건물의 지하실에서 다른 케블들과 같이 묶어 진다. 이로부터 큰 케블묶음이 중앙사무소의 지하실을 지나게 된다.

또한 배선실은 위성과 지구국의 련결 혹은 지점 대 지점의 지구미크로파련결가운데서 어느 하나를 미크로파안테나에 련결하는 기능을 수행한다. 안테나에 의한 접속은 전용망에 속하는 경우도 있지만 AT&T 또는 MCI와 같은 장거리반송파에 접속하기 위한 국부적인 수단으로도 될수 있다.

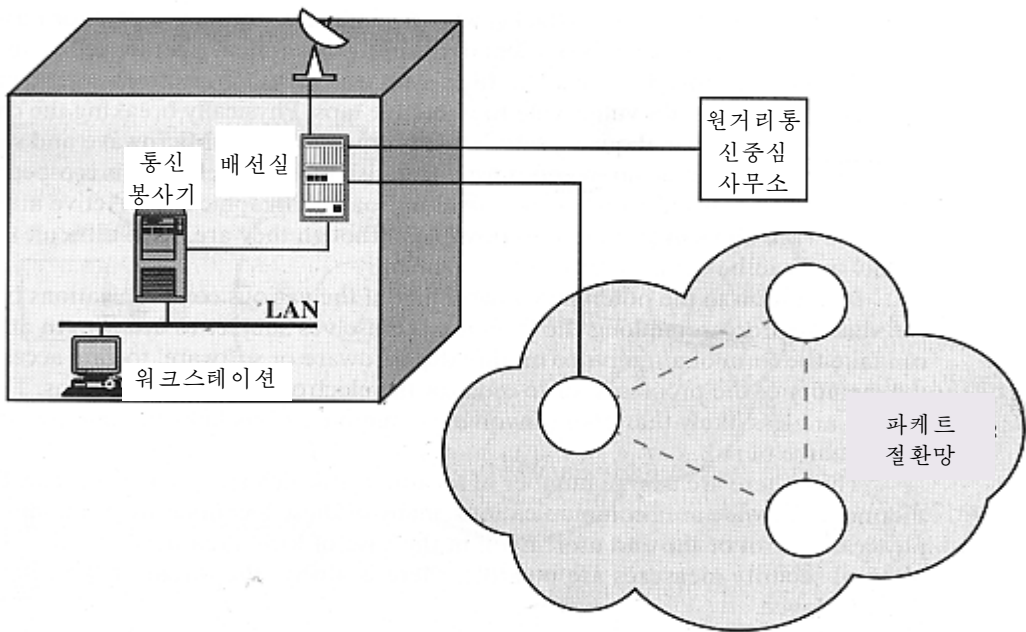



그림 5-1. 공격당할 지점

배선실은 파케트절환망의 마디에 대한 접속을 제공할수도 있다. 이 접속은 임대회선이나 직접적인 전형회선일수도 있고 ISDN과 같은 공개원격통신망을 통한 절환접속일수도 있다. 망의 내부에서 자료는 목적지말단체계가 접속된 마디에 도착할 때까지 많은 마디들과 마디들사이의 련결을 통과한다.

공격은 임의의 통신련결에서 일어 날수 있다. 적극적공격에서 공격자는 련결통로에 대한 물리적조종을 필요로 하며 그를 통해 통신정보를 포착하려고 한다. 소극적공격에서 공격자는 단순히 통신자료를 알려고 한다. 통신련결수단으로는 케블(쌍심전화선, 동축케블, 빛섬유), 미크로파련결, 위성통신로들이 포함된다. 쌍심전화선과 동축케블은 도청장치 혹은 전자기파를 감시하는 감응장치를 리용하여 공격할수 있다. 접촉형도청장치(Invasive tap)는 능동적공격과 소극적공격을 다 할수 있으나 감응형도청장치는 흔히 소극적인 공격만을 할수 있다. 빛섬유인 경우는 도청수단을 적용할수 없는데 이것이 이 매체의 우점이다. 빛섬유는 전자기마당을 발생시키지 않으므로 감응형도청장치의 적용이 불가능하다. 케블의 물리적파괴는 신호의 질에 심

각한 영향을 미치며 곧 검출된다. 따라서 접촉형도청도 불가능하다. 미크로파나 위성통신에서 능동적인 공격이 기술적으로 어렵고 비용이 들지만 역시 가능하다.

각이한 통신련결우에서 잠재적약점들외에 그 통신로에 따르는 여러 처리기들자체도 공격을 받는다. 공격은 하드웨어나 쏘프트웨어를 변경시키는 형태를 취하며 그 처리기의 기억기접근을 얻거나 전자기발신을 감시할수 있다. 이 공격들은 통신접속의 경우보다는 드물지만 여전히 위험한것들로 되고 있다.

즉 공격이 일어 날수 있는 위치는 대단히 많다. 더우기 광대역통신에서 많은 위치들은 말단사용자의 물리적통제를 받지 않는다. 물리적보안대책이 가능한 국부망의 경우에도 거기에는 나쁜 의도를 가진 직원이 늘 있을수 있다.

련결에서의 암호화와 말단 대 말단에서의 암호화

앞절에서 본것처럼 위협에 대처하는 가장 강력하며 공통적인 방법은 암호화이다. 암호를 공격방어에 리용하는데서는 우선 암호화하고 암호기의 설치위치를 결정해야 한다. 그림 5-2에 지적한것처럼 두가지 방안이 있을수 있다. 즉 련결(Link)에서의 암호화와 말단 대 말단에서의 암호화이다.

기본방법

련결암호화에서 위험한 매 통신련결에는 그의 량끝점에 어떤 암호장치가 설치된다. 그리하여 전체 통신련결상에서 모든 전송들은 비밀로 된다. 큰 망의 경우 이 방법에서는 많은 암호장치들이 요구되지만 그 의의는 크다. 이 방법의 결함의 하나는 파케트절환에 들어 갈 때마다 그 통보문을 복호하여야 하는것인데 그것은 그 파케트를 경로조종하기 위하여 스위치가 파케트의 머리부에서 주소(가상회선번호)를 읽어야 하기때문이다. 따라서 통보문은 매 스위치에서 공격 받을수 있다. 만일 공개파케트절환망에서 작업한다면 사용자는 마디들의 보안에 대한 아무런 조종도 할수 없다.

련결암호와 관련된 몇가지 문제들을 지적한다. 이 전략이 효과적으로 되기 위해서는 원천지로부터 목적지에 이르는 모든 가능한 련결들에서 련결암호를 사용하여야 한다. 어떤 련결을 공유하는 마디들의 모든 쌍은 유일한 열쇠를 공유하여야 하며 매 련결들에서는 다른 열쇠가 리용된다. 그러므로 많은 열쇠들이 준비되여야 한다. 여기서 매 열쇠는 다만 2개의 마디에만 배송되여야 한다.

말단 대 말단암호화에서는 암호화처리가 두 말단체계에서 진행된다. 원천지의 가입자 또는 말단은 자료를 암호화한다. 암호화된 형식의 자료는 목적말단 혹은 가입자에게 망을 통하여 변경없이 전송된다. 목적지는 원천지와 열쇠를 공유하며 따라서 자료를 복호할수 있다. 이 방안에서는 망련결 혹은 절환들에 대한 공격을 막고 전송의 보안을 보장할수 있다. 따라서 말단 대 말단암호화는 그 통신을 지원하는 망과 련결의 보안 정도에 관여하는 말단사용자를 믿는다. 그러나 여기에도 약점이 있다.

다음과 같은 상황을 생각하자. 한 가입자가 어떤 X.25파케트절환망에 접속하고 다른 가입자에 가상회선을 설정하며 말단 대 말단암호화를 리용하여 자료를 전송하려고 한다고 하자. 이때 자료는 머리부와 사용자자료로 이루어 진 파케트형식으로 망에서 전송된다고 하자. 매 파케트의 어느 부분을 가입자가 암호화하겠는가? 가입자가 머리부를 포함하여 전체 파케트를 암호화한다고 가정하자. 파케트절환마디는 암호화된 파케트를 받고 그 머리부를 읽을수 없다. 그러므로 그 파케트를 보낼수 없게 된다. 그로부터 가입자는 파케트의 사용자자료부분만 암호화하고 머리부는 그대로 남겨야 한다는것을 알수 있다.

그러므로 말단 대 말단암호화로 사용자자료는 안전하다. 그러나 전송패턴은 그렇지 못한데 그것은 파케트의 머리부들이 그대로 전송되기때문이다. 한편 말단 대 말단암호화는 인증기능을 가지고 있다. 만일 말단체계가 어떤 암호열쇠를 공유하면 확인자는 관계되는 열쇠를 송신자만이 공유하였으므로 짐작되는 송신자로부터 통보문을 수신하였음을 확증할수 있다. 그러한 인증이 련결암호방식에는 없다.

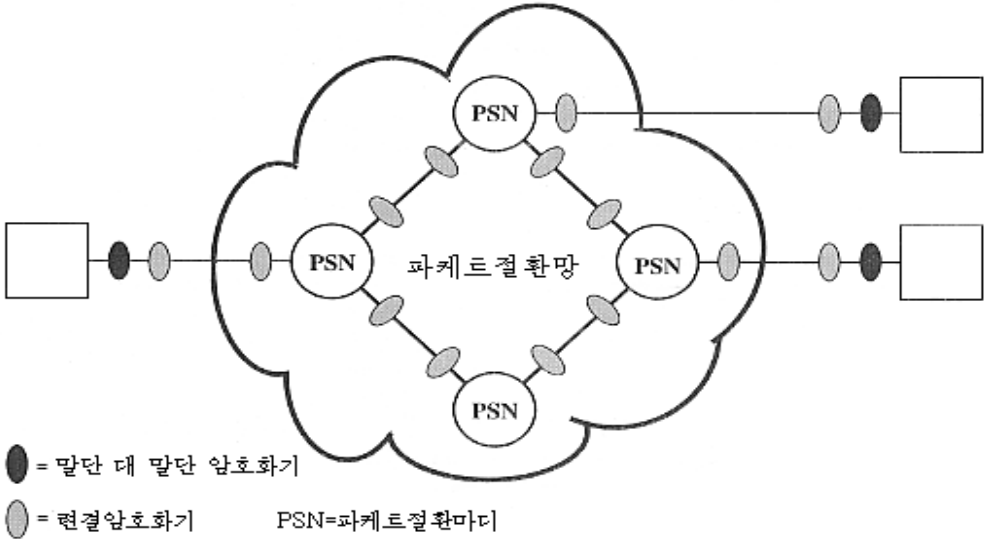



그림 5-2. 파케트절환망에서의 암호화

표 5-1. 련결과 말단 대 말단에서의 암호화의 특성[PFLE97]

련결암호	**말단 대 말단암호**
말단체계와 중개체계내에서의 보안	
송신자가 분석한 통보문	송신자가 암호화한 통보문
중간마디들에서 로출된 통보문	중개마디들에서 암호화된 통보문
사용자의 역할	
송신자에 의하여 적용	송신처리에 의하여 적용
사용자에게 투명	사용자는 암호화를 적용한다.
가입자는 암호화수단을 보존한다.	사용자는 알고리듬을 결정하여야 한다.
모든 사용자들에 대하여 하나의 수단	사용자는 암호방식을 선택한다.
하드웨어로 실현할수 있다.	쏘프트웨어실현
통보문이 모두 암호화되거나 일부만 암호화될수 있다.	사용자는 매 통보문에 대하여 암호화하겠는가 안하겠는가를 결정한다.
실현에 관련된것	
가입자중개마디당 하나의 열쇠를 요구하며 중간의 마디쌍에 대해서도 하나의 열쇠가 요구된다.	매 사용자쌍마다 하나의 열쇠를 요구한다.
가입자에 인증기능을 제공한다.	사용자인증을 제공한다.

보다 강도 높은 보안을 위하여서는 그림 5-2에 보여 준것처럼 련결과 말단 대 말단의 두가지 암호법을 다 적용하는것이 필요하다. 암호화방식이 다 리용되면 다음 말단 대 말단암호화열쇠를 리용하여 파케트의 사용자자료부분을 암호화한다. 전체 파케트는 련결암호화열쇠를 리용하여 암호화된다. 파케트가 망을 통하여 전송되면 매 절환기는 머리부를 읽기 위해 련결암호열쇠를 리용하여 파케트를 복호한다. 그리고 그것을 다음련결에 보내기 위하여 다시 파케트의 전체를 암호화한다. 전체 파케트는 그 파케트가 어떤 파케트스위치의 기억기에 실제로 있는 동안을 제외하고는 안전하다.

표 5-1은 두 암호화전략들의 열쇠특징들을 개괄하였다.

말단 대 말단암호화기능의 론리적배치

련결암호화에서 암호화기능은 통신계층의 낮은 준위에서 수행된다. OSI의 방식으로 접속암호화는 물리층 혹은 련결층에서 진행된다.

말단 대 말단암호화를 위한 암호기능의 론리적배치에 대한 여러가지 선택이 가능하다. 가장 낮은 실천적준위로서 암호화기능은 망층에서 실현될수 있다. 실례로 암호는 X.25와 결합될수 있으며 따라서 모든 X.25파케트들의 사용자자료부분이 암호화된다.

망층암호화에서 확인가능하고 개별적으로 보호되는 개체들이 그 망의 말단체계들의 수에 대응된다. 매 말단체계는 다른 말단체계와 만일 그 두 체계가 비밀열쇠를 공유하였다면 암호화된것을 교환하도록 맞물릴수 있다. 매 말단체계내에서 모든 사용자처리와 응용들은 특정의 목적말단체계에 도달하기 위하여 같은 열쇠와 같은 암호방식을 리용할수 있다. 이러한 약속으로 암호기능을 어떤 앞단처리기(일반적으로 말단체계에서 통신접속대)에 제기할수도 있다.

그림 5-3에 앞단처리기(FEP)의 암호화기능을 보여 준다. 가입자측에서 FEP는 파케트를 접수한다. 파케트의 사용자자료부분은 암호화되며 한편 파케트머리부는 암호화과정을 건너 뛴다. 결과 파케트는 망에 넘겨 진다. 반대방향으로 망으로부터 도착하는 파케트들에 대하여 사용자자료부분이 복호되며 전체 파케트는 가입자에 넘겨 진다. 만일 전송층의 기능이(즉 ISO전송 규약 또는 TCP) 앞단에서 실현된다면 전송층머리부는 그대로 남게 되며 전송규약자료단위의 사용자자료부분은 암호화된다.

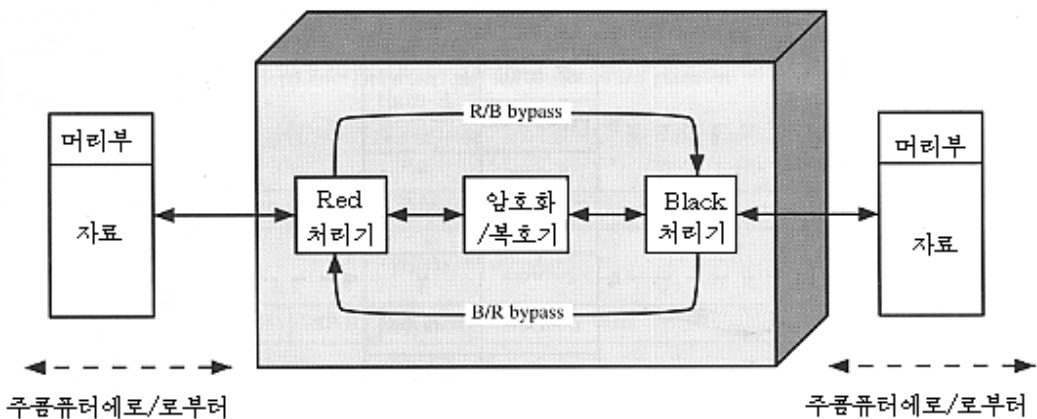



그림 5-3. 앞단처리기기능

망층 X.25나 TCP와 같은 말단 대 말단규5약에 관한 암호봉사의 개발은 완전히 통합된 호상련결망내에서 거래에 대한 말단 대 말단보안을 제공한다. 그러나 그러한 방식은 전자우편, 전자자료교환(EDI), 파일전송과 같은 호상련결망의 경계들을 지나는 전송에 필요한 봉사를 제공할수 없다.

그림 5-4에 그것을 보여 준다. 이 실례에서 전자우편관문은 TCP/IP에 기초한 방식과 OSI에 기초한 방식을 리용하는 호상련결망을 접속하는데 리용되였다. 그러한 구성에서는 응용층밑의 말단 대 말단규약이 없다. 매 말단체계로부터의 전송층과 망층과의 련결은 전자우편관문에서 진행되며 다른 말단체계에 접속하기 위한 새로운 전송층과 망접속들이 설정된다. 더우기 그러한 씨나리오는 두개의 서로 다른 방식들사이의 관문경우로 제한되지 않는다. 지어 두 말단체계가 TCP/IP 또는 OSI을 리용한다면 다른 고립된 호상접속망들사이에 전자우편관문들이 설치되는 실지 구성방식들의 많은 실례들이 있게 된다. 축적전송기능을 가지는 전자우편과 같은 응용들에서 말단 대 말단암호화를 위한 장소는 응용층뿐이다.

응용층암호화의 약점은 고찰해야 할 실체의 수가 극적으로 늘어 난다는것이다. 수백개의 가입자들을 지원하는 망은 수천의 사용자 또는 처리들을 지원할수 있다. 따라서 더 많은 비밀열쇠를 생성하여 배송하여야 한다.

흥미 있는 다른 중요한 방법의 하나는 통신층에 의하여 암호화할 정보량은 적지만 안전성은 더 높아 진다는것이다. 그림 5-5에서 이것을 TCP/IP방식을 실례 들어 강조하였다. 그림에서 응용준위관문은 응용준위에서 조작되는 축적전송장치를 말한다.

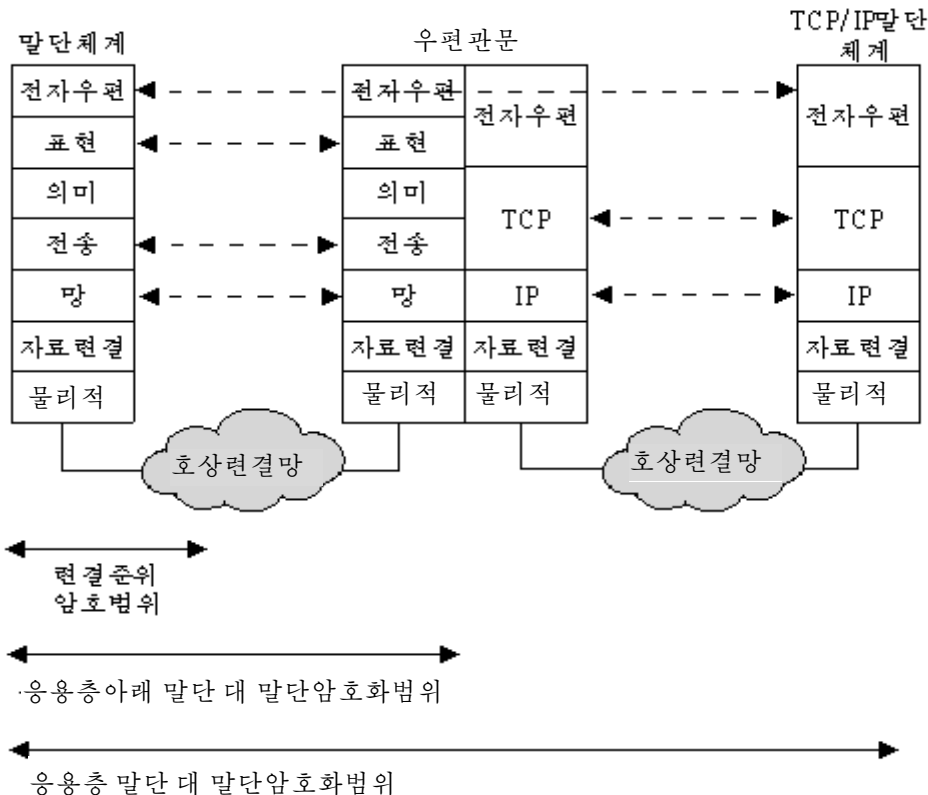



그림 5-4. 축적회송통신들사이 암호화적용범위

응용준위암호화에서(그림 5-5의 ㄱ) TCP부분의 사용자자료부만 암호화된다. TCP, IP, 망준위, 접속준위머리부, 접속준위꼬리부는 암호화되지 않는다. 반면에 암호화가 TCP준위(그림 5-5의 ㄴ)에서 진행된다면 단일말단 대 말단접속에서 사용자자료와 TCP머리부는 암호화된다. 원천지로부터 목적지까지 IP데이터그램을 경로조종하여야 하므로 IP머리부는 그대로 남는다. 그러나 어떤 통보문이 관문을 지난다면 TCP접속은 종결되며 새로운 전송접속이 다음 전송단계를 위해 개방된다. 더우기 관문은 IP의 기초화에 의해서 목적지로 취급된다. 따라서 자료단위의 암호화된 부분은 관문에서 부호화된다. 만일 다음 단계가 TCP/IP망우의것이라면 사용자자료와 TCP머리부는 전송전에 다시 암호화된다. 그러나 관문 그자체에서 자료단위는 평문으로 취급된다. 마지막으로 접속준위암호화에서(그림 5-5의 ㄷ) 접속머리부와 꼬리부를 제외한 전체 자료단위는 매 접속에서 암호화되지만 전체 자료단위는 매 경로조종기와 관문에서는 그대로 된다.

Link-H	Net-H	IP-H	TCP-H	Data	Link-T

ㄱ) 응용준위암호화(련결, 경로기, 관문우에서)

Link-H	Net-H	IP-H	TCP-H	Data	Link-T

련결들과 경로조종기들에서

Link-H	Net-H	IP-H	TCP-H	Data	Link-T

관문들에서

ㄴ) TCP준위암호화

Link-H	Net-H	IP-H	TCP-H	Data	Link-T

련결들에서

Link-H	Net-H	IP-H	TCP-H	Data	Link-T

경로조종기들과 관문들에서

ㄷ) 련결준위암호화

TCP-H : TCP머리부
IP-H : IP머리부
Net-H : 망준위머리부(실레로 X.25 파케트머리부, LLC머리부)
Link-H : D자료련결조종규약머리부
Link-T : D자료련결조종규약꼬리부

그림 5-5. 각이한 암호전략들의 관계

5.2 전송기밀성

1장에서 일부 경우에 사용자들이 전송분석으로부터 보안에 대하여 관심을 둔다는것을 언급하였다. 마디들사이의 통보문들의 길이와 개수에 대한 정보는 적이 누가 누구에게 통신하고 있는가를 결정할수 있게 한다 .이것은 군사분야에서 더 명백하다. 지어 상업응용들에서 정보분석에 의해 전송의 발송자가 감추려 하는 정보를 얻을수 있다. 문헌[MUFT89]에 전송분석공격으로부터 로출될수 있는 다음과 같은 정보들을 소개하였다.

- 전송대방들의 식별자
- 대방들의 통신회수
- 중요한 정보가 교환된다고 추측되는 통보문패턴, 통보문길이 또는 통보문의 량
- 대방들사이의 특정한 대화와 관련한 사건들

전송에 관계되는 다른 문제는 비밀통로를 창조하기 위한 전송패턴의 리용이다. 비밀통신로는 통신수단의 설계자가 의도하지 않은 방식에서의 통신수단이다. 일반적으로 통신로를 통한 정보전송에서는 보안방략이 위반될수 있다. 실례로 어떤 직원은 관리자에게 발견되지 않으면서 쉽게 도청할수 있는 방법으로 외부와 정보를 통신하려고 할수 있다. 쌍방은 어떤 길이보다 작은 명백히 합법적인 통보문은 2진수 0을 표시하고 긴 통보문은 1을 표시하는 부호를 설정할수 있다.

련결암호화수법

련결암호화에서는 전송정보분석의 기회를 줄이기 위해 파케트머리부를 암호화한다. 그러나 그런 환경에서도 여전히 공격자는 망에서의 전송자료량에 접근할수 있으며 매 말단체계에서 나가고 들어 가는 전송량을 감시할수 있는 가능성이 있게 된다. 이러한 공격에 대한 효과적인 대책은 그림 5-6에 제시한 전송의 메꾸기이다.

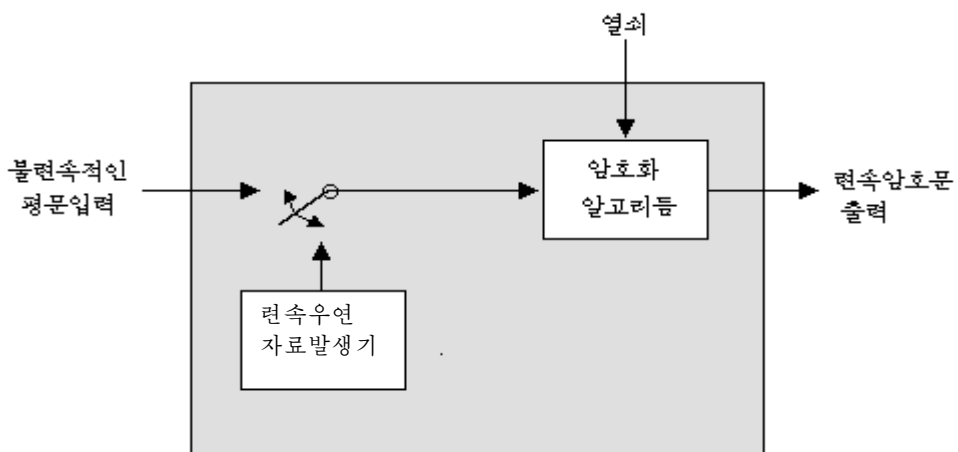



그림 5-6. 전송메꾸기암호화장치

전송의 메꾸기는 비록 평문이 주어 지지 않더라도 련속적으로 암호문출력을 생성한다. 입력평문이 없을 때에는 암호문이 전송된다.이것은 공격자가 진짜 자료와 메꾸기사이를 구분할수 없게 하며 결국 전송량을 추측할수 없게 한다.

말단 대 말단암호수법

전송메꾸기는 본질적으로 련결암호화기능이다. 만일 말단 대 말단암호화가 사용되면 방어에 써먹을수 있는 수단은 더 제한된다. 실례로 암호화가 응용층에서 실현된다면 적은 어느 전송실체가 전송에 참가하는가를 결정할수 있다. 만일 암호화기술이 전송층에서 적용되면 망층주소와 전송패턴은 접근가능하게 된다.

여기서 전송층이나 응용층에서 자료단위들을 같은 길이로 채워 넣는것이 효과적인 기술이다. 또한 빈 통보문은 임의로 정보흐름에 삽입될수 있다. 이러한 전술은 적이 말단사용자들사이의 자료교환량을 알수 없게 하며 기본전송패턴을 애매하게 만든다.

5.3 열쇠배포

전통암호를 리용하자면 교환쌍방이 같은 열쇠를 공유하여야 하는데 이 열쇠는 다른 사람으로부터 엄격히 보호되여야 한다. 더우기 공격자가 열쇠를 알수 있다면 빈번히 열쇠를 바꾸어야 한다. 모든 암호체계의 강도는 **열쇠배포기술**의 완성을 전제로 한다. 두 대방 A, B에 대한 열쇠배포는 다음과 같은 방법으로 실현될수 있다.

1. A의 열쇠는 A가 선정하고 물리적으로 B에 배송한다.
2. 3자가 열쇠를 선정하여 물리적으로 A와 B에 배송한다.
3. 만일 A와 B가 이전과 최근에 하나의 열쇠를 리용하였다면 한 대방은 낡은 열쇠로 새 열쇠를 암호화하여 다른 대방에게 배송할수 있다.
4. A와 B가 각각 제3자 C에 대한 암호화접속을 가지면 C는 A와 B에 암호화된 련결들에로 열쇠를 보낼수 있다.

경우 1과 2를 열쇠의 수동적배송(Manual delivery)이라고 부른다. 련결암호화에서 그것은 매 련결암호화장치가 그 련결의 대방하고만 자료교환을 하기때문에 정당한 요구로 된다. 그러나 말단 대 말단암호화에서는 수동적배송이 곤난하다. 분산체계에서 이미 주어 진 가입자나 말단은 다른 많은 가입자나 말단과 정보교환을 위해 서로 련계를 가져야 한다. 따라서 매 장치에는 동적으로 많은 열쇠들이 공급되여야 한다. 이 문제는 광대역분산체계의 경우 특히 더 어렵다.

문제의 규모는 지원하여야 할 통신쌍들의 수에 관계된다. 만일 말단 대 말단암호화가 망 혹은 IP준위에서 주어 지면 열쇠는 통신하려는 망우의 매 가입자쌍만큼 필요된다. N개의 가입자에 요구되는 열쇠수는 $[N(N-1)]/2$이다. 만일 암호화가 응용준위에서 진행되면 열쇠는 통신을 요구하는 모든 사용자쌍 혹은 처리사이에 요구된다. 따라서 망이 수백개의 가입자들을 가지면 수천의 사용자들과 처리들이 있게 된다. 그림 5-7에 말단 대 말단암호화를 위한 열쇠배포과제의 크기를 보여 준다. 1000개 마디를 가지는 마디준위암호화를 리용하는 망에서 50만개의 열쇄배송을 생각할수 있다. 만일 이 망이 1만개의 응용을

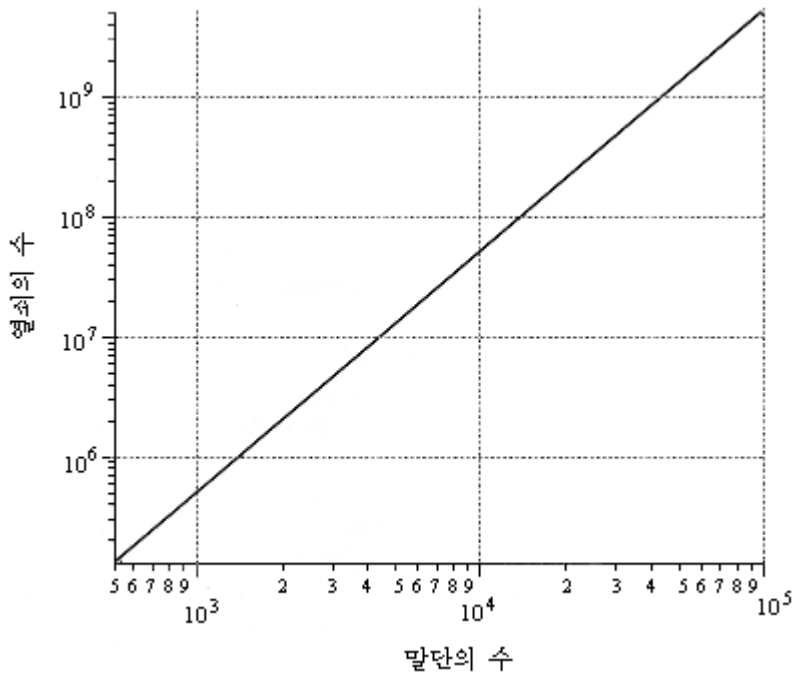



그림 5-7. 말단들사이의 임의의 접속을 지원하는데 요구되는 열쇠의 수

지원한다면 응용준위의 암호화에 5천만개의 열쇠가 요구될것이다.

우의 열쇠배포에서 3번째 경우를 보자. 이것은 련결암호화나 말단 대 말단암호화에서 가능하다. 만일 공격자가 한개의 열쇠에 대한 접근에 성공한다면 뒤따르는 열쇠들도 모두 로출될것이다. 더우기 잠재적으로 수백만개의 열쇠의 초기배송이 되여 있어야 한다.

말단 대 말단암호화에서 우의 4번째 안의 변종이 많이 쓰이고 있다. 이 방식에서 열쇠배포쎈터는 요구가 제기될 때마다 사용자(가입자, 처리, 응용)쌍에 배송한다. 매 사용자는 열쇠를 배정 받기 위하여 열쇠배포쎈터와 한개의 열쇠를 공유하여야 한다.

열쇠배포쎈터의 리용은 열쇠의 계층적리용에 기초하였다. 최소한 두개 준위의 열쇠가 리용되고 있다(그림 5-8). 말단체계들사이의 통신은 보통 대화(Session)열쇠라고 부르는 림시열쇠를 리용하여 암호화된다. 일반적으로 대화열쇠는 가상회로 또는 전송접속과 같은 론리적접속기간에만 리용되고 그다음에는 페기된다. 매 대화열쇠는 말단사용자통신에 리용되는 같은 망수단들상에서 열쇠배포쎈터로부터 얻어 진다. 따라서 대화열쇠는 열쇠배포쎈터와 말단체계 혹은 사용자가 공유하는 **주열쇠**를 리용하여 암호화된 모양으로 전송된다.

매 말단체계 혹은 사용자들에게는 열쇠배포쎈터와 공유하는 유일한 주열쇠가 있다. 물론 주열쇠들도 어떤 방법으로 배송되여야 한다. 그러나 문제의 규모는 매우 축소된다. 만일 거기에 통신을 요구하는 N개의 실체가 있다면 매번 $[N(N-1)]/2$개의 대화열쇠가 요구될것이다. 그러나 매 실체에 대하여 다만 N개의 주열쇠가 요구된다. 따라서 주열쇠들은 물리적배송과 같은 비암호적방법으로 배송할수 있다.

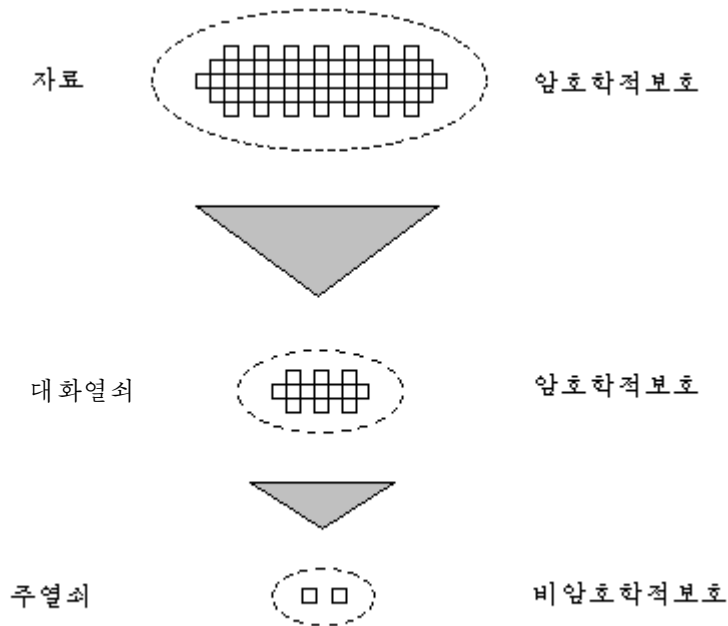



그림 5-8. 열쇠계층의 리용

열쇠배포씨나리오

열쇠배포의 개념은 여러가지 방법으로 정의할수 있다. 전형적인 씨나리오를 그림 5-9에 보여 준다. 이 씨나리오는 열쇠배포쎈터(KDC)에 의하여 매 사용자가 유일한 주열쇠를 가지게 된다고 가정한다.

사용자 A가 B와 론리적접속을 설정하려고 한다고 보고 그 접속우에서 전송된 자료를 보호하기 위하여 1회용(한번 쓰고 버리는)대화열쇠를 요구한다고 하자. A는 비밀열쇠 K_a를 가진다. 그것은 다만 A와 KDC만 안다고 하자. 마찬가지로 B는 KDC와 함께 주열쇠 K_b를 가지고 있다. 다음과 같은 절차들이 있을수 있다.

1. A는 B에 대한 론리적접속을 보호하기 위한 대화열쇠를 KDC에 요구한다. 통보문에는 A와 B의 신원(이름)과 한번쓰기정보(nonce)라고 부르는 이 트랜잭션을 위한 유일한 식별자 N_1가 포함된다. 한번쓰기정보로는 시간도장(timestamp), 계수기 또는 란수가 될수 있는데 최소한 그것을 매 요청에 대하여 구별할수 있어야 한다. 또 가장을 하지 못하도록 한번쓰기정보에 대하여 적이 추측하기 힘들게 하여야 한다. 따라서 란수는 한번쓰기정보로서 좋은 방법이다.
2. KDC는 K_a를 리용하여 암호화된 통보문으로 응답한다. 따라서 A만이 그 통보문을 성과적으로 받을수 있으며 A는 그것이 KDC에서 보낸것이라는것을 알수 있다. 그 통보문에는 A에 대한 다음과 같은 2개의 항목이 포함된다.

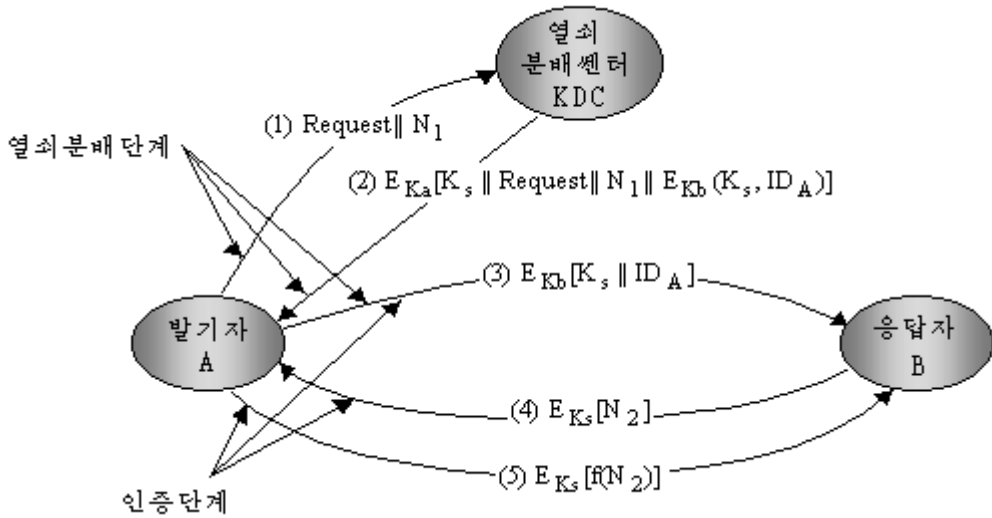



그림 5-9. 열쇠배포씨나리오

¤ 해당 대화에서 리용되는 1회용대화열쇠 K_S

¤ 한번쓰기정보를 포함하는 초기요청통보문. 이것은 A가 해당한 요청과 이 응답을 대조할수 있게 한다.

따라서 A는 그의 초기요청이 KDC에 의하여 접수되기전에 변경되지 않았다는것과 어떤 선행한 요청의 재연이 아니라는것을 한번쓰기정보를 리용하여 확인할수 있다. 그외에 통보문에는 B에 대한 다음의 2개의 항목이 포함된다.

¤ 해당 대화에서 리용되는 1회용대화열쇠 K_S

¤ A의 식별자(그것의 망주소), ID_A

이 마지막 두 항목들은 K_b로(KDC와 B가 공유하는 주열쇠) 암호화된다. 이것들을 B에 보내여 접속을 실현하며 A의 신원을 확인한다.

3. A는 해당 대화에 필요한 대화열쇠를 보관하고 KDC에서 발생한 정보 $E_{kb}[K_S\|ID_A]$를 B에게 보낸다. 이 정보는 K_b를 리용하여 암호화되였기때문에 도청하지 못한다. 대화열쇠(K_S)를 알면 대방이 A라는것(ID_A로부터)과 KDC에서 보낸 정보임을 알게 된다(E_{kb}를 리용하여 암호화하였기때문에).

이 시점에서 대화열쇠는 A와 B에 안전하게 배송되며 A와 B는 보호된 정보교환을 시작할수 있게 된다. 그러나 2개의 보충적인 단계가 필요하다.

4. 암호화를 위해 새롭게 만들어 낸 대화열쇠를 리용하여 B는 A에게 한번쓰기정보 N_2을 보낸다.
5. A는 K_S를 리용하여 $F(N_2)$로 응답한다. 여기서 F는 N_2에 대하여 어떤 변환을 수행하는 함수이다.

이 단계들은 B가 재연이 없이 받은 처음 통보문이라는것을 보증한다. 실제적인 열쇠배포는 단계 1~3에서 진행되며 단계 4, 5는 인증기능을 수행한다.

계층적열쇠조종

하나의 KDC로 열쇠배포기능을 제한할 필요는 없다. 대단히 큰 망에서 그렇게 하는것은 실천적이 못된다. 따라서 다른 하나의 방안은 KDC의 계층을 설정하는것이다. 실례로 단일국부망 혹은 단일건물과 같은 총적인 호상 접속망의 작은 령역을 담당한 국부 KDC들이 있을수 있다. 동일한 국부령역안에서 실체들사이의 통신을 위하여 국부 KDC는 열쇠배포를 책임진다. 만일 다른 령역의 두 실체가 열쇠공유를 바란다면 대응하는 국부 KDC들은 대역 KDC를 통하여 통신할수 있다. 이 경우에 포함된 세 KDC중의 임의의 하나는 실제로 열쇠를 선택할수 있다. 계층개념은 망의 지리학적범위와 사용자집단의 크기에 따라 3 혹은 그 이상의 층으로 확장할수 있다.

계층방식은 주열쇠의 배송에 필요한 비용을 최소화하는데 그것은 대부분의 주열쇠들이 국부 KDC에 의하여 그의 국부실체들과 공유되기때문이다. 그밖에 그러한 방식은 고장 또는 파괴된 KDC로 인한 피해를 그의 국부지역만으로 제한한다.

대화열쇠의 생명주기

대화열쇠들이 자주 교체될수록 그것들은 더욱 안전해 진다. 그것은 적이 임의의 주어 진 대화열쇠에 대한 암호문을 보다 적게 가지기때문이다. 한편 대화열쇠의 배송은 정보교환을 지연시키며 망용량에 부담을 준다. 보안관리자는 개별적대화열쇠의 생명주기를 결정하는데서 이 모순되는 두 문제의 균형을 맞추어야 한다.

접속지향규약에서 한가지 명백한 방법은 매개 새로운 대화에 대하여 새로운 대화열쇠를 리용하며 접속이 개방된 시간에 대해서는 같은 대화열쇠를 리용하는것이다. 만일 론리적접속이 매우 긴 생명주기를 가진다면 대화열쇠를 PDU(규약자료단위)렬번호가 순환할 때마다 주기적으로 변경하는것이 중요하다.

트랜잭션규약과 같은 비접속규약에 대하여 명백한 접속의 시작과 종결이 없다. 그러므로 얼마나 자주 대화열쇠를 변경시켜야 하는가가 명백치 않다. 가장 안전한 수법은 매 교환에 대하여 새로운 대화열쇠를 사용하는것이다. 그러나 이것은 매개의 트랜잭션에 대한 지연과 간접비용을 최소로 하는 접속규약의 원리적인 우점의 하나에 모순된다. 보다 좋은 전략은 어떤 고정된 주기동안만 혹은 어떤 일정한 수의 트랜잭션들에만 주어 진 대화열쇠를 리용하는것이다.

투명한 열쇠조종방식

그림 5-9에서 제시한 방법에는 많은 변종들이 있는데 그중 하나를 여기서 취급한다. 그림 5-10에 준 방식은 말단사용자들에게 투명한 방법으로 망 혹은 전송준위에서 말단 대 말단암호화를 제공하는데 쓸모 있다. 이 방법은 통신이 X.25 또는 TCP와 같은 접속지향말단 대 말단규약을 리용한다는것을 전제로 한다. 이 방법의 주목되는 부분은 가입자 또는 말단을 대신하여 대화열쇠를 얻고 말단 대 말단암호화를 수행하는 앞단처리기(그림 5-3에서 보여 준것과 같은)이다.

그림 5-10에 접속실현과정의 단계를 보여 준다. 한 가입자가 다른 가입자와 접속을 요청한다면 접속요청파케트를 보낸다(단계1). 앞단처리기는 그 파케트를 보관하고 접속의 확립을 허락 받기 위하여 KDC에 요청한다(단계2). FEP와 KDC사이의 통신은 KDC와 FEP만 공유하는 주열쇠를 리용하여 암호화된다. 만일 KDC가 접속요청을 승인하면 그것은 대화열쇠를 발생하고 매 앞단처리기에 유일한 불변열쇠를 리용하여 두 해당 앞단처리기들에 대화열쇠를 보낸다(단계3). 요청된 앞단처리기는 접속요청파케트를 내보내며

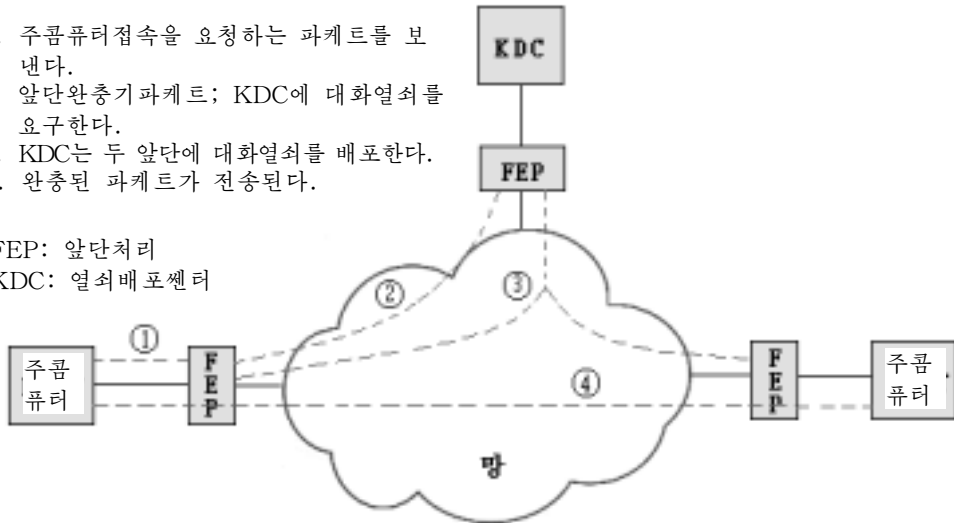



그림 5-10. 접속지향규약에서 자동열쇠배포

두 말단체계사이의 접속이 설정된다(단계4). 두 말단체계사이에서 교환되는 모든 사용자자료는 1회용대화열쇠를 리용하여 이 개별적앞단처리기들에 의하여 암호화된다.

이 수법의 우점은 말단체계들에 대한 영향을 최소화한다는것이다. 가입자의 견지에서 보면 FEP는 파케트절환마디처럼 보이며 망에 대한 가입자의 대면부는 변화되지 않는다. 망의 견지에서 보면 FEP는 가입자처럼 보이며 그 가입자에 대한 파케트절환대면부는 변하지 않는다.

분산열쇠조종

열쇠배포쎈터의 리용은 KDC가 파괴로부터 보호되고 또 신임되여야 한다는 요구를 제기한다. 이 요구는 만일 열쇠배포가 완전히 분산화된다면 피할수 있다. 비록 완전분산화가 전통암호만을 사용하는 큰 망에서는 실천적이 못되더라도 어떤 국부적인 정황에서는 쓸모 있을수 있다.

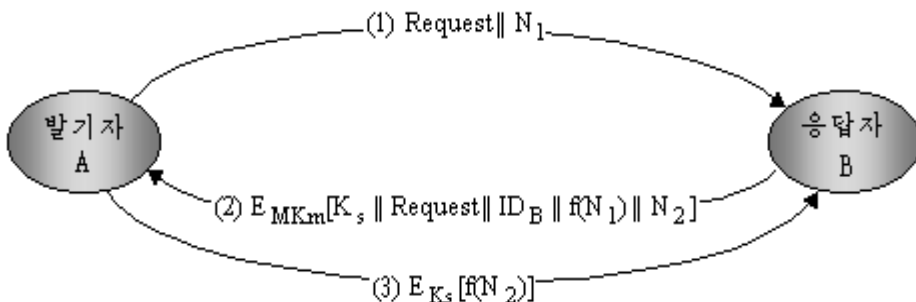



그림 5-11. 분산화된 열쇠배포

분산화방법은 매 말단체계가 대화열쇠배포를 위하여 모든 가능한 대방말단체계들과 안전한 방법으로 통신할수 있을것을 요구한다. 따라서 n개의 말단체계를 가지는 구성에서는 $[n(n-1)]/2$개의 주열쇠를 필요로 할수 있다.

대화열쇠는 다음과 같은 단계로 설정될수 있다(그림 5-11).

1. A는 한번쓰기정보 N_1를 포함하는 대화열쇠에 대한 요구를 B에 보낸다.
2. B는 공유한 주열쇠를 리용하여 암호화된 통보문으로 응답한다. 이 응답은 B에 의하여 선정된 대화열쇠, B의 식별자, 값 $f(N_1)$, 다른 한번쓰기정보 N_2을 포함한다.
3. A는 새 대화열쇠를 리용하여 B에 $f(N_2)$을 보낸다.

따라서 매 마디점은 기껏 $(n-1)$개의 주열쇠들을 관리하지만 필요되는만큼 대화열쇠들을 생성하여 리용할수 있다. 주열쇠를 리용하여 전송되는 통보문이 짧기때문에 암호분석이 어렵다. 앞에서와 같이 보호를 목적으로 하여 대화열쇠는 제한된 시간동안만 사용된다.

열쇠사용의 조종

열쇠계층의 개념과 자동열쇠배포기술의 리용은 수동적으로 관리하고 배송하여야 할 열쇠의 개수를 많이 감소시켰다. 자동적으로 배송되는 열쇠들을 리용하는 방법에 대하여 일부 조종을 부가할 필요가 있을수도 있다. 실례로 대화열쇠로부터 주열쇠를 구분하는 외에 다음과 같은 사용에 기초하여 대화열쇠들의 서로 다른 형태들을 정의할수 있다.

- 망에서 일반적인 통신을 위한 자료암호화열쇠
- 전자송금이나 상업거래에서 쓰이는 개인식별번호(Personal identification number: PIN)용의 PIN암호화열쇠
- 공개적으로 접근할수 있는 지점에 보관된 파일들을 암호화하는 파일암호화열쇠

형태별에 따르는 열쇠구분의 가치를 보기 위하여 주열쇠가 자료암호화열쇠로서 장치에 들어 갈수 있는 위험성을 고찰하자. 일반적으로 주열쇠는 말단체계와 열쇠배포쎈터의 암호장치내에 물리적으로 보안되여 있다.

이 주열쇠로 암호화된 대화열쇠는 그러한 대화열쇠로 암호화된 자료와 마찬가지로 응용프로그람에서 리용할수 있다. 그러나 주열쇠가 대화열쇠로 취급되면 인증되지 않은 응용에서 그 주열쇠로 암호화된 대화열쇠들의 평문을 얻을수 있다.

그러므로 이 열쇠들과 관련된 특성에 기초하여 열쇠들의 리용방법을 제한하는 체계의 조종을 제정하는것이 좋을것이다. 하나의 간단한 방안은 어떤 타그를 매 열쇠와 결합시키는것이다. 제안된 기술은 DES에서 리용할수 있으며 매 64-bitDES열쇠에서 여분의 8bit를 리용하게 한다. 즉 기우성검사를 위하여 예약된 8개의 열쇠비트들은 열쇠타그를 이룬다. 이 비트들은 다음과 같이 해석된다.

- 한 비트는 그 열쇠가 주열쇠인가 대화열쇠인가를 지적한다.
- 한 비트는 그 열쇠를 암호화에 리용할수 있는가 없는가를 지적한다.
- 한 비트는 그 열쇠가 복호화에 리용될수 있는가 없는가를 지적한다.
- 나머지비트들은 앞으로의 리용을 예견하여 남겨 놓는다.

타그가 열쇠에 매몰되여 있으므로 열쇠가 배송되면 그 열쇠와 함께 암호화되며 결국 보호되게 된다. 이 방식에서 결함은 첫째로, 그 타그의 길이가 8bit로 제한되므로 기능성과 유연성에 제한을 주며 둘째로, 타그가 명백한 형으로 전송되지 않으므로 복호화시점에서 리용만 할수 있으며 열쇠리용을 조종하는 방법에서 제한을 준다는것이다.

조종벡토르라고 부르는 더 유연한 방식이 문헌[MATY9]에 소개되였다.이 방식에서 매 대화열쇠는 그것을 제한하고 그 리용을 지적하는 많은 마당들로 이루어 지는 조종벡토르와 결합된다. 조종벡토르의 길이는 가변이다.

조종벡토르는 KDC에서 열쇠발생시에 암호학적으로 열쇠와 결합된다. 결합과 해체를 그림 5-12에 보여 준다. 첫 단계로 조종벡토르는 길이가 암호열쇠의 길이와 같은 값을 주는 하쉬함수를 거치게 된다. 하쉬함수는 8장에서 구체적으로 취급한다. 본질적으로 하쉬함수는 큰 범위로부터 비교적 균일하게 분포되는 작은 범위에로의 넘기기이다. 실례로 1∼100범위의 수들이 1∼10범위의 수들로 하쉬되면 원천값의 약 10%가 매개 목표값에로 넘기게 된다.

다음 하쉬값은 대화열쇠를 암호화하기 위한 열쇠입력으로서 리용되는 출력을 생성하기 위하여 주열쇠와 XOR된다. 즉

$$\text{하쉬값} = H = h(CV)$$
$$\text{열쇠입력} = K_m \oplus H$$
$$\text{암호문} = E_{km \oplus H}[K_S]$$

여기서 k_m은 주열쇠, k_S는 대화열쇠이다. 대화열쇠는 다음 거꿀연산에 의하여 평문으로부터 얻을수 있다.

$$K_S = P_{km\oplus H}[\,E_{km\oplus H}[K_S]]$$

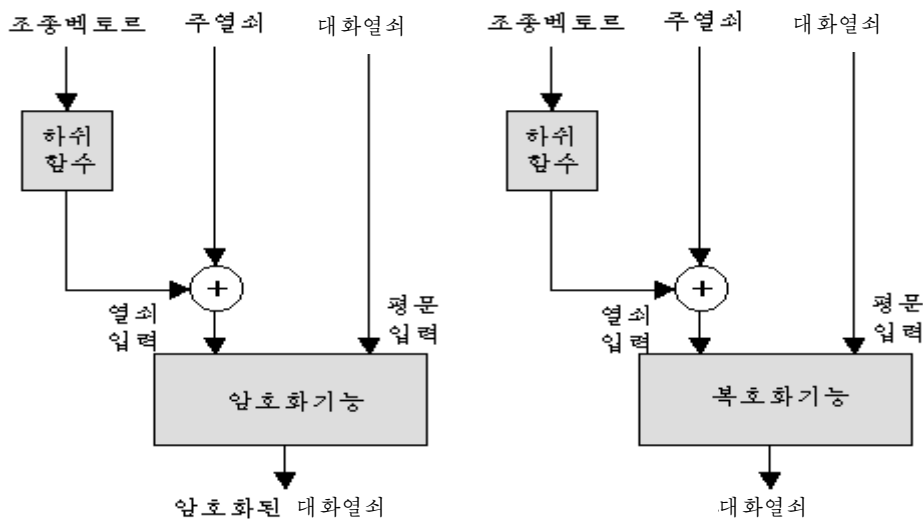



그림 5-12. 조종벡토르암호화와 복호화

대화열쇠가 KDC로부터 사용자에게 배송될 때 그것은 그대로의 조종벡토르와 함께 배송된다. 대화열쇠는 사용자가 KDC와 공유하는 주열쇠와 조종벡토르를 리용하여서만 회복할수 있다. 따라서 대화열쇠와 그의 조종벡토르사이의 관계는 유지된다.

조종벡토르의 리용은 8-bit타그의 리용에 비해 두가지 우점을 가진다. 첫째로, 조종벡토르의 길이에 제한이 없는데 그것은 열쇠리용에서 임의의 복잡한 조종을 할수 있게 한다. 둘째로, 조종벡토르는 연산의 모든 단계에서 그대로 쓸수 있다는것이다. 따라서 열쇠의 리용에 대한 조종을 여러 위치에서 실시할수 있다.

5.4 란수발생

란수는 각이한 망보안응용에서 암호화를 리용할 때 중요한 역할을 논다. 이 절에서는 망보안에서의 란수리용의 기본적인 개괄을 주고 다음에 란수생성의 몇가지 방법을 고찰한다.

란수의 리용

암호에 기초한 많은 망보안알고리듬들은 란수를 리용한다. 실례로

- 그림 5-9와 5-11에서 보여 준것과 같은 호상인증방식들. 이 두 열쇠배포씨나리오에서 한번쓰기정보들은 공격의 재연을 방지하기 위한 핸드쉐이킹(Handshaking)을 위하여 리용된다. 한번쓰기정보에서 란수를 리용하면 한번쓰기정보를 추측하거나 결정하려는 적의 기도를 좌절시킨다.
- KDC 혹은 통신자에 의해서 수행되는 대화열쇠생성, 대화열쇠생성이 KDC에 의하여 진행되는가 아니면 본인에 의하여 진행되는가.
- RSA공개열쇠암호알고리듬의 열쇠생성(6장)

이러한 응용은 란수렬에 대하여 두개의 명백하나 꼭 량립하지 않는 성질 즉 무질서성과 예측불가능성을 요구한다.

우연성

전통적으로 란수렬의 발생에서 기본은 수렬이 어떤 잘 정의된 통계적의미에서 무질서하게 되는것이였다. 수렬의 우연성을 평가하기 위하여 다음 두 기준이 쓰이고 있다.

- 평등분포(Uniform distrlbution) : 렬에서 수의 분포는 평등하여야 한다. 즉 매개 수의 출현빈도는 근사적으로 같아야 한다.
- 독립성: 렬에서 한 수의 값은 다른것들의 영향을 받지 않는다.

어떤 수렬이 평등분포와 같은 특정의 분포에 따르는가에 대한 검사법은 잘 정의되여 있으나 독립성을 증명하는 검사법은 없다. 오히려 어떤 수렬의 독립성이 보장되지 않는다는것을 보여 주는데 적용할수 있는 검사법들이 있다. 일반적인 전략은 독립성이 충분히 강하게 존재한다는 확신이 얻어 질 때까지 그러한 검사들을 많이 하는것이다.

앞으로의 론의에서는 통계학적으로 무질서한 수렬이 암호에 관계되는 알고리듬설계에서 많이 쓰인다. 실례로 6장에서 론의되는 공개열쇠암호방식의 기본요구는 씨수발생가

능성이다. 일반적으로 주어 진 큰수 N이 씨수인가를 결정하는것은 어렵다. 힘내기공격은 N을 $\sqrt{N}$ 보다 작은 모든 옹근수들로 나누는것이다.

만일 N이 차수로서 10^{150}이고 공개열쇠암호에서 흔히 나타나는 옹근수라면 그러한 전수공격방법은 사람이나 콤퓨터의 능력을 초월한다. 그러나 수의 씨수성을 판정할수 있는 많은 효과적인 알고리듬들이 있는데 거기서는 입력으로서 우연적으로 선택된 옹근수의 렬을 리용하여 비교적 단순한 계산으로 씨수성을 판정한다. 렬이 충분히 길다면(그러나 물론 $\sqrt{10^{150}}$ 보다는 훨씬 작다.) 수의 씨수성은 거의 정확하게 결정할수 있다. 이러한 방법을 우연화방법이라고 부르는데 알고리듬설계에서 빈번히 나타난다. 요컨대 어떤 문제가 너무 어렵거나 그 문제를 정확히 푸는데 시간이 너무 걸린다면 우연화방법에 기초한 보다 단순하고 간결한 이러한 방법이 임의의 요구하는 수준의 확신성을 가진 답을 제공하는데 리용된다.

비예측성

대화열쇠발생, 호상인증과 같은 응용에서 제기되는 요구는 수렬이 통계학적으로 무질서할뿐아니라 수렬에서 다음 출현할 수를 예측할수 없어야 한다는것이다.

《진짜》우연렬에서 매수는 그 수렬에서 다른 수들과 통계학적으로 독립이며 따라서 예측불가능하다. 그러나 간단히 말한다면 진짜 란수는 거의 리용되지 않으며 어떤 알고리듬에 의하여 생성된 무질서한것처럼 보이는 수렬이 자주 쓰인다. 이 알고리듬에 의하여 발생된 렬은 고유한 의미에서 독립이 아니므로 적이 먼저 출현하는 수렬의 부분에 기초하여 그 다음에 출현할 수들을 예측할수 없게 하는데 신중한 주의를 돌려야 한다.

란수의 원천

진짜 란수의 원천은 얻기 힘들다. 방사선복사의 이온화에 의한 임플스검출기, 가스배출관과 루실축전기(leaky capacitor)같은 물리적잡음발생기들이 원천으로 될수 있다. 그러나 그러한 장치들은 망보안응용의 리용에서 제한을 가진다. 여기에는 두가지 문제 즉 무질서성(우연성)과 이러한 수들의 정확성에 대한것이 있게 된다[BRIG79].

그러나 이러한 란수는 망보안응용에서 제기되는 란수의 량에 대한 요구를 충족시킬수 없다. 더우기 서고들에 있는 란수들이 진짜로 통계적으로 무질서하다고 해도 그것은 그 서고의 리용을 알고 있는 적이 복사물을 얻을수 있기때문에 예측할수 있다.

따라서 암호응용에서는 일반적으로 생성을 위해 알고리듬적기술을 리용한다. 이 알고리듬은 결정적이며 따라서 통계학적으로 무질서하지 않은 수렬을 생성한다. 그러나 알고리듬이 훌륭하다면 얻어 진 수렬은 무질서성에 대한 많은 합법적인 검사에 통과될것이다. 그러한 수들을 흔히 모조란수(Pseudorandom number)라고 부른다.

결정론적알고리듬에 의하여 생성되는 수를 란수로서 리용하는데서 다소 확신이 없을수 있다. 그러한 문제들에 대한 철학적의미에서 론의가 많으나 실천에서는 사용하고 있다.

이 개념은 확률론의 전문범위로서 문헌 [HAMM91]에 소개되여 있다.

모조란수발생기

현재 널리 쓰이고 있는 모조란수발생기와는 상당히 다르지만 레머(Lehmer)가 처음으로 제안한 선형합동식방법이라고 부르는 알고리듬을 보자. 알고리듬은 4개의 파라메터에 의하여 결정되는데 다음과 같다.

m	모드	$m>0$
a	배수	$0 \le a < m$
c	중분	$0 \le c < m$
X_0	출발값 또는 씨	$0 \le X_0 < m$

준우연렬 $\{X_n\}$는 다음과 같은 점화방정식에 의하여 얻어 진다.

$$X_{n+1} = (aX_n + c) \bmod m$$

만일 m, a, c, X_0이 옹근수이면 이 방법은 매 옹근수가 $0 \le X_n \le m$범위에 놓이는 수렬을 생성할것이다.

a, c, m값의 선택은 좋은 란수발생기개발에서 관건적이다. 실례로 $a=c=1$이라고 하자. 그러면 생성되는 렬은 명백히 나쁘다. 다시 $a=7$, $c=0$, $m=32$, $X_0=1$이라고 하자. 발생된 렬은 {7,17,23,1,7,…}인데 질이 나쁘다. 모든 가능한 32개의 값중에서 겨우 4개만 나타난다. 수렬의 주기는 4이다. 만일 $a=5$로 바꾸면 수렬은 {1, 5, 25, 29, 17, 21, 9, 13, 1…} 즉 주기가 8로 된다.

m이 대단히 크게 되면 긴 우연렬을 생성할수 있다. m선택에서 공통적인 기준은 주어 진 콤퓨터에서 표현가능한 최대 부아닌 옹근수와 거의 같게 하는것이다. 즉 m은 거의 2^{31}에 가깝게 정해 진다.

모조란수발생기(선형합론법)의 설계에서 쓰이는 세 기준은 다음과 같다[PARK88].

T_1: 함수는 완전주기발생함수로 되여야 할것이다. 즉 함수는 0～m사이의 모든 수들을 반복없이 발생하여야 한다.

T_2: 발생된 렬은 무질서하여야 한다. 물론 렬은 결정론적으로 발생시키였으므로 무질서하지 않다. 그러나 렬이 무질서성을 판정하는 여러 검사에서 합격하여야 한다.

T_3: 함수는 32-bit산술로 효률적으로 실현되여야 한다.

적당한 a, c, m를 가지고 우의 세 기준을 만족시킬수 있다. 기준 T_1에 대하여 m은 씨수이고 $c=0$이면 a의 어떤 값에 대하여 생성되는 렬의 주기는 $m-1$로 된다. 다만 값 0은 제외된다. 32-bit산술을 위해 m의 값은 보통 $2^{31}-1$로 정해 진다. 이때 생성관계식(함수)은 다음과 같다.

$$X_{n+1} = (aX_n) \bmod (2^{31} - 1)$$

a의 선택에서는 $0 \sim 2^{31}-1$범위의 자연수가 가능하지만 우의 세 기준을 만족시키는 점은 얼마 안된다. 그러한 a의 하나가 $7^5=16807$이다. 이것은 IBM363계렬에서 리용되였다. 이 발생기는 광범히 리용되였고 다른 란수발생기보다 검사도 더 많이 받았다. 이 란수는

통계학적모의작업에 자주 쓰이고 있다.

선형합동식란수의 강도는 a와 m이 적당히 선택되면 결과의 수렬이 모임 1, 2, …, m−1로부터 우연적으로 뽑아 낸 렬과 통계적으로 식별불가능할것이다. 초기값 X_0의 선택과는 별도로 알고리듬에 대하여 완전히 우연적인것은 없다. 일단 그 값이 선택되면 나머지 수들은 결정된다. 만일 적이 선형합동식알고리듬이 리용되였다는것과 그의 파라메터를 아는 경우(례로 a=7^5, c=0, m=2^{31}−1) 하나의 수를 알게 되면 모든 뒤따르는 수를 알수 있게 된다. 적이 다만 선형합동식란수알고리듬이 리용되였다는것만 안다면 렬의 극히 작은 부분에 대한 지식만이면 그 알고리듬의 파라메터를 결정하는데 충분하다. 적이 수렬의 값 X_0, X_1, X_2, X_3을 안다고 하자. 그러면

$$X_1 = (aX_0 + c)\bmod m$$
$$X_2 = (aX_1 + c)\bmod m$$
$$X_3 = (aX_2 + c)\bmod m$$

이 방정식으로부터 a, c m을 구할수 있다.

그러므로 비록 좋은 모조란수발생기라고 하여도 비예측성이 좋을것이 요구되는데 이것은 수렬의 어떤 부분에 대한 지식도 적이 수렬의 그 이후의 원소들을 결정하는데 불충분하여야 한다는것이다. 이러한 목표는 여러가지 방법으로 달성할수 있다. 실례로 란수렬을 변경시키는 내부체계시계를 리용할수 있다[BRIG79]. 이 시계를 리용하는 하나의 방법은 새로운 씨로 현재의 시간값(mod m)을 리용하여 매 N개의 수다음에 수렬을 재시작하는것이고 다른 방법은 매 란수(mod m)에 현재의 시간값을 단순히 첨가하는것이다.

암호학적으로 발생되는 란수

암호응용을 위한 란수를 생성하기 위해 암호론리의 우점을 써먹을수 있다. 많은 방법들이 리용되는데 여기서는 3가지 실례를 제시한다.

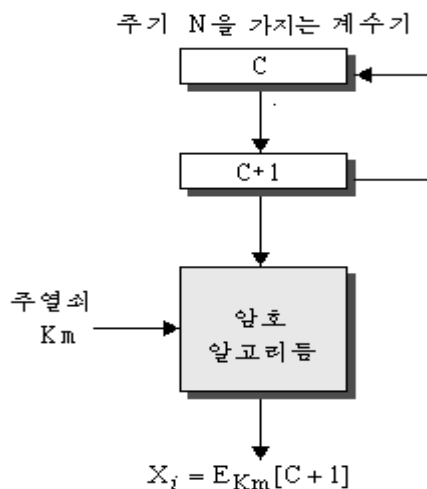



그림 5-13. 계수기에 의한 모조란수발생

순환암호화

그림 5-13에 문헌 [MEYE82]에서 제안된 방법을 보여 주었다. 이 경우에 절차는 주열쇠로부터 대화열쇠를 발생시키는데 리용된다. 주기가 N인 계수기에 의해 암호론리에 대한 입력이 제공된다. 실례로 만일 56-bitDES열쇠가 생성되면 주기 2^{56}을 가진 계수기가 사용될수 있다. 매 열쇠가 생성된 후에 계수기는 1만큼 증가한다. 이 도식에 의하여 생성된 모조란수들은 전체 주기에 걸쳐 순환한다. 출력 $X_0, X_1, \wedge X_{N-1}$의 매개는 서로 다른 계수기값에 기초하였으며 따라서 $X_0 \neq X_1 \neq \wedge \neq X_{N-1}$이다. 주열쇠가 보호되므로 초기의 하나 또는 여러개의 열쇠값들에 대한 지식을 가지고는 어떤 비밀열쇠도 계산론적으로 추론해 낼수 없다.

알고리듬의 강도를 더 높이기 위하여 단순한 계수기가 아니라 완전주기모조란수발생기의 출력을 입력으로 할수 있다.

DES출력반결합방식

DES의 출력반결합방식(OFB)을 그림 3-14에 보여 주는데 그것은 흐름암호에서처럼 열쇠발생을 위해서도 사용할수 있다. 조작의 매 단계의 출력은 64-bit값이고 암호화를 위해 맨 왼쪽 j개 비트들이 반결합된다는것에 주목하자. 련속되는 64-bit출력은 좋은 통계적성질을 가지는 모조란수렬을 생성한다. 앞의 소제목에서 제안된 방법에서와 마찬가지로 보호된 주열쇠를 리용하면 발생된 대화열쇠들을 보호한다.

ANSI X9.17모조란수발생기

암호학적으로 가장 강한 모조란수발생기의 하나를 ANSI X9.17에서 지적하였다. 재정보안응용과 PGP를 비롯한 많은 응용들에서 이 기술이 리용된다.

그림 5-14에 암호화를 위하여 3중DES를 리용한 알고리듬을 보여 준다. 구성은 다음과 같다.

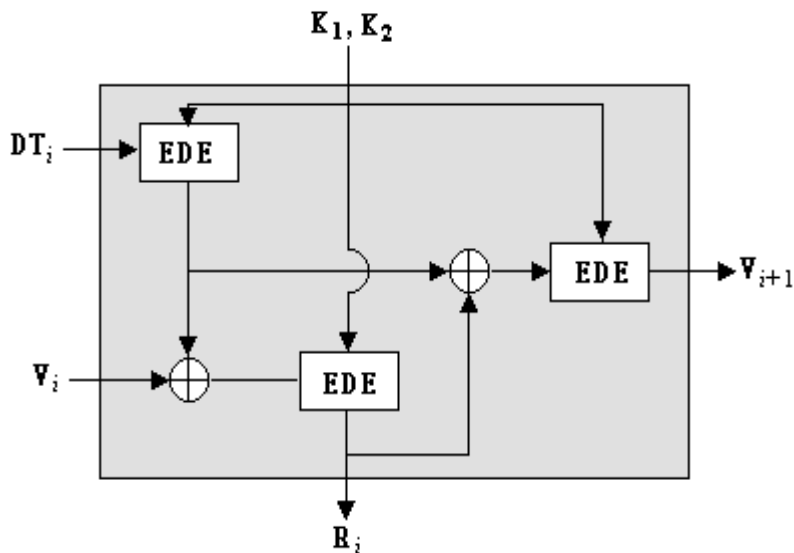



그림 5-14. ANSI X9.17모조란수발생기

- 입력: 두 모조란수입력은 발생기를 구동한다. 하나는 현재의 날자와 시간을 표현하는 64bit인데 매 발생때마다 갱신된다. 다른 하나의 입력은 64-bit씨값인데 어떤 임의의 값으로 초기화되고 발생과정에 갱신된다.
- 열쇠: 발생기는 3개의 3중DES암호모듈들을 리용한다. 그 3개는 다 같이 56-bit열쇠의 쌍을 리용하는데 그것들은 안전하게 보관되여야 하며 모조란수발생을 위해서만 리용된다.
- 출력: 출력은 64-bit모조란수와 64-bit씨값이다.

다음과 같은 량들을 약속한다.

DT_i: i번째 발생단계의 초기의 날자/시간값

V_i: i번째 발생단계의 시작에서의 씨값

R_i: i번째 발생단계에서 생성되는 모조란수

K_1, K_2: 매 단계에서 리용되는 DES열쇠들

그러면

$$R_i = \mathrm{EDE}_{K_1,K_2}[V_i \oplus \mathrm{EDE}_{K_1,K_2}[DT_i]]$$
$$V_{i+1} = \mathrm{EDE}_{K_1,K_2}[V_i \oplus \mathrm{EDE}_{K_1,K_2}[DT_i]]$$

여기서 EDE는 두열쇠3중DES를 리용하는 암호-복호-암호렬이다.

몇가지 요인들이 이 방법의 암호강도에 영향을 준다. 112-bit열쇠와 9개의 DES암호화를 위한 3개의 EDE암호화를 포함한다. 이 방식은 2개의 준우연입력에 의하여 구동되는데 그것은 날자 및 시간값과 그 발생기에 의하여 생성되는 모조란수와 다른 발생기에 의하여 생성된 씨이다. 따라서 적이 취급하여야 할 자료의 량은 매우 방대해 진다. 란수 R_i가 의심되더라도 R_i로부터 V_{i+1}를 추론해 내는것은 V_{i+1}를 생성하는데 EDE연산이 보충적으로 리용되였으므로 불가능하다.

BBS 발생기

안전한 모조란수의 발생에 대한 가장 통속적인 방법은 BBS발생기로 알려 진 방법이다[BLUM86]. 그것은 암호학적강도에 대한 가장 강한 공개적증명인것 같다. 절차는 다음과 같다. 첫째로, 두개의 큰 씨수 p와 q를 선택하는데 그것들은 모두 4로 나누면 나머지가 3인 씨수이다. 즉

$$p \equiv q \equiv 3 (\mathrm{mod}\, 4)$$

이 표기는 7장에서 더 구체적으로 설명되는데 그것은 (p mod4)＝(q mod4)＝3을 의미한다. 실례로 씨수 7과 11은 $7 \equiv 11 \equiv 3 (\mathrm{mod}\, 4)$를 만족시킨다. $n = p \times q$라고 하자. 다음으로 n과 서로 소인 란수 s를 하나 선택한다. 이것은 p와 q가 s의 인수가 아니라는 것과 동등하다. BBS발생기는 다음 알고리듬에 의하여 비트들의 렬 B_i를 생성한다. 즉

$$X_0 = s^2 \bmod n$$
for $i=1$ to ∞
$$X_i = (X_{i-1})^2 \bmod n$$
$$B_i = X_i \bmod 2$$

표 5-2. BBS발생기의 조작실례

s	X_i	B_i
0	20749	
1	143135	1
2	177671	1
3	97048	0
4	89992	0
5	174051	1
6	80649	1
7	45663	1
8	69442	0
9	186894	0
10	177046	0

s	X_i	B_i
11	137922	0
12	123175	1
13	8630	0
14	114386	0
15	14863	1
16	133015	1
17	106065	1
18	45870	0
19	137171	1
20	48060	0

따라서 매 반복에서 제일 아래자리비트가 취해 진다. 표 5-2에 BBS조작의 실례를 주었다. 여기서 n=192649=383×503, s=101355이다.

BBS는 **암호학적으로 안전한 준우연비트발생기**(CSPRBG)로서 취급되였다. CSPRBG는 **다음-비트검사**를 통과하는것으로 정의되는데 그 정의는 다음과 같다[MENE97]:《준우연비트발생기는 출력렬의 첫 k개 비트의 입력에 대하여 ½보다 더 큰 확률로 (k+1)번째 비트를 예측할수 있는 다항식시간알고리듬이 존재하지 않으면 다음-비트검사를 통과한다고 말한다.》 다른 말로 그 렬의 첫 k개의 비트들이 주어 지면 ½보다 큰 확률을 가지고 다음 비트가 0 혹은 1인가를 결정하는 실천적알고리듬이 없다는것이다. 모든 실천적목적을 위하여 렬들은 예측불가능하여야 한다. BBS의 보안은 n의 인수분해의 곤난성에 기초한다. 즉 n이 주어 지면 그의 두 씨인수 p, q를 결정하여야 한다(문헌[STIN95]).

참고문헌

BRIG79 Bright, H., and Enison, R. “Quasi-Random Number Sequences from Long-Period TLP Generator with Remarks on Application to Cryptography.” *Computing Surveys*,December 1979.

EAST94 Eastlake, D.; Crocker, S.; and Schiller, J. *Randomness Recommendations for Security.* RFC 1750, December 1994.

FUMY93 Fumy, S., and Landrock, P. “Principles of Key Management.” *IEEE Journal on Selected Areas in Communications,* June 1995.

KNUT98 Knuth, D. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms.*

Reading, MA: Addison-Wesley, 1998
MENE97 Menezes, A.; Oorshcot, P.; and Vanstone, S. *Handbook of Applied Cryptography* Boca Raton, FL: CRC Press, 1997.
MEYE82 Meyer, C., and Matyas, S. *Cryptography: A New Dimension in Computer Data Security*. New York: Wiley, 1982
STIN95 Stinson, D. *Cryptography: Theory and Practice*. Boca Ration, FL: CRC Press, 1995
ZENG91 Zeng. K.; Yang, C.; Wei, D.; and Rao, T. "Pseudorandom Bit Generators in Stream-Cipher Cryptography." *Computer*, February 1991.

문 제

1. 전자우편체계는 여러 수신자들을 취급하는 방법에 따라 차이난다. 어떤 체계에서는 우편발송자가 모든 필요한 복사를 할수 있으며 이것들을 필요한 수신인에게 독립적으로 보낸다. 다른 방법은 매 수신인에게 보내기 위한 경로를 먼저 결정하는것이다. 그다음 하나의 통보문은 그 경로의 공통부분에 송신되고 복사물들은 그 경로가 갈라 질 때에만 생성된다. 이러한 처리를 우편배깅(bagging: 우편자루에 넣기)이라고 부른다.
   ㄱ) 보안측면을 무시하고 두 방법의 상대적인 우점과 결점을 지적하시오.
   ㄴ) 두 방법의 보안에 관한 요구조건과 문제점에 대하여 설명하시오.
2. 5.2에서는 비밀통로를 구축하는 방법으로서 통보문길이를 리용하는 방법을 주었다. 비밀통신로를 구축하기 위하여 전송패턴을 리용하는 보충적인 방식을 설명하시오.
3. 한 LAN관리자는 그림 5-15에 제시된것과 같은 열쇠배포수단을 제공한다.
   ㄱ) 그 방식을 서술하시오.
   ㄴ) 그림 5-9의 방식과 비교하시오. 찬성과 반대되는 점은 무엇인가.
4. c=0인 선형합동식알고리듬 즉

$$X_{n+1} = (aX_n) \bmod m$$

   을 취할 때 m이 씨수이고 a의 주어 진 값이 m−1과 같은 최대주기를 준다면 a^k 역시 최대주기를 주게 되며 k는 m보다 작고 m−1은 k로 완제되지 않는다는것을 밝힐수 있다.
   X_0=1, m=31을 리용하여 a=3, 3^2, 3^3, 3^4에 대한 렬을 만들어 보시오.
5. 다음의 생성기로부터 얻을수 있는 최대주기는 얼마인가.

$$X_{n+1} = (aX_n) \bmod 2^4$$

   ㄱ) a의 값은 얼마인가.
   ㄴ) 씨에 부과되는 제한은 무엇인가.
6. mod 2^{31}은 보충비트가 없이 표시할수 있어 모드연산을 더 쉽게 수행하기때문에 선형합동식법에서 mod 2^{31}대신에 mod 2^{31}−1을 선택하는것이 이상할수 있다. 일반적으로 mod 2^k−1이 mod 2^k보다 더 적합하다. 왜 그런가.

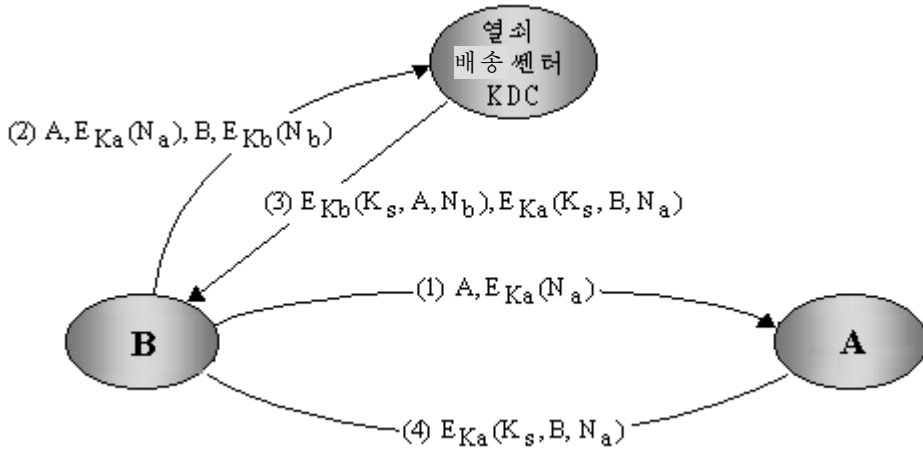



그림 5-15. 문제 5-3을 위한 그림

7. 선형합동식알고리듬에서 완전주기를 가지게끔 파라메터를 선택하는것이 꼭 좋은 무질서성을 주는것은 아니다. 실례로 다음 발생기를 고찰하시오.

$$X_{n+1} = (6X_n)\,\mathrm{mod}13$$
$$X_{n+1} = (7X_n)\,\mathrm{mod}13$$

두 렬의 완전주기를 쓰시오. 어느것이 더 무질서한것 같은가?

8. 암호, 모의 또는 통계적설계 등과 같은 모조란수의 리용에 의거하고 있는 콤퓨터체계서고의 란수발생기를 무턱대고 믿는것은 위험하다. 연구자들은 많은 최신도서들과 프로그람파케트들에서 모조란수발생에 좋지 못한 알고리듬들을 리용하고 있다는것을 지적하고 있다. 이 련습도 현재의 콤퓨터체계를 검토하게 한다.검토는 우연적으로 선택된 두 옹근수들의 최대공약수가 1일 확률이 $6/\pi^2$과 같다는 씨저(Cesaro)의 정리에 기초하였다.
π의 값을 통계적으로 결정하기 위하여 프로그람에서는 이 정리를 리용한다. 주프로그람은 3개의 부분프로그람들 즉 체계서고로부터 우연옹근수를 발생시키기 위한 란수발생기, 유클리드련제법을 리용하여 두 옹근수의 최대공약수를 계산하는 부분프로그람, 뿌리를 계산하는 부분프로그람을 호출하여야 한다. 만일 마지막프로그람들이 없으면 자신이 작성하여야 할것이다. 주프로그람은 앞에서 언급한 확률을 평가하기 위하여 많은 란수들을 다 순환할것이다. 이로부터 자기가 π를 결정하는 편이 간단하다.
그 결과가 5.14에 가깝다면 성공이다. 만일 그렇지 않으면 결과는 아마도 2.7 정도의 낮은 값으로 될것이다. 왜 그러한 나쁜 결과가 얻어 지는가?

9. 쌍방이 같은 비밀열쇠를 소유하였는가를 확인하는 다음방법을 가정하자. 일방은 열쇠길이의 우연비트렬을 만들어 그것을 그 열쇠로 XOR한다. 그리고 결과를 통로에 내보낸다. 한편 대방은 자기의 열쇠(상대방의 열쇠와 같아야 한다.)로 들어 오는 블로크를 XOR하고 그것을 다시 통로에 내보낸다. 일방은

자기가 받은것이 처음우연렬과 같은가를 검사하고 대방이 같은 비밀열쇠를 가지고 있다는것 또는 전송된 열쇠를 아직도 못가지고 있다는것을 확인한다. 이 방식의 부족점은 무엇인가?

10. 《우리는 지금 매우 곤난한 환경에 처해 있네, 홈스. 런던의 한 외국대사관의 여러 콤퓨터에 민감한 정부문서의 복사물이 기억되였다는 정보를 입수했네. 흔히 이 문서들은 매우 엄격한 보안요구를 만족시키는 선택된 몇개의 정부콤퓨터에만 전자문서의 형태로 보관되고 있거든. 그러나 때때로 그것들은 모든 정부콤퓨터들을 접속하는 망을 통하여 보내야 할 때가 있단 말이요. 이 망에서 모든 통보문은 가장 권위 있는 암호전문가들에 의하여 보증된 고급암호알고리듬을 리용하여 암호화되였소.그러니 NSA나 KGB로도 그것을 파괴할수 없네. 그런데 지금 그 문서가 어떤 나라의 한 외교관의 손에 들어 갔거든.참 어떻게 했으면 좋겠나.》

《누군지 짚히는데가 없는가요?》

《글쎄, 조사를 좀 해봤는데 …다만 지금 누가 정부콤퓨터에 대한 합법적접근권한을 가지고 있으며 누가 그 대사관의 외교관과 빈번히 접촉하는가를 알고 있기는 한데 그 콤퓨터는 문서들이 표준적으로 기억되여 있는 콤퓨터도 아니란 말이요.그런데 그가 어떻게 그 문서의 복사품을 얻었는지 참 수수께끼거든. 어쨌든 그는 그것을 복호하지 못할거요.》

《그 망에서 쓰는 통신규약은 어떤것인가요.》

《대체로 이렇네. 망의 매 마디는 부과된 유일한 비밀열쇠 K_n을 가지는데 그 열쇠는 마디와 신용되는 봉사기사이의 비밀통신에 리용되지. 즉 모든 열쇠들은 다 그 봉사기에 기억된단 말이요. 사용자 A가 사용자 B에 비밀통보문 M을 보려고 한다면 다음 규약으로 시작할수 있네. 즉

1. 란수 R를 발생하고 수신측에 자기의 이름 A, 목적지 B 그리고 $EK_n[R]$을 보낸다.
2. 봉사기는 A에 $EK_b[R]$로 응답한다.
3. B에 $EK_b[R]$와 $E_R[M]$을 보낸다.
4. B는 K_b를 알며 $EK_b[R]$를 복호하여 R를 얻으며 R에 의해 $E_R[M]$을 복호하여 M을 얻는다.

통보문을 보낼 때마다 란수가 발생되여야 한다는걸 당신도 알고 있을거요. 나는 극비신용마디사이에서 통보문이 도청될수 있었다고 보네. 그러나 아마 복호는 못할거요.》

《그럴수도 있지. 규약은 비밀이 아닐게고… 봉사기가 요청을 보내는 사용자를 인증하지 않을테니까… 분명히 규약의 설계자들은 X(와 봉사기)가 K_x를 알고 있을 때에나 $EK_x[R]$를 보내는것이 사용자 X를 송신자라고 암시적으로 인증한다고 본것 같네. 그러나 당신도 알다싶이 $EK_x[R]$가 도청되여 후에 재연될수 있단 말이요. 그 구멍의 위치만 알면 그가 접근할수 있는 콤퓨터를 통해 사람들을 감시하면서 충분한 증거를 얻을수 있을것이 아닌가. 모름직이 그는 이렇게 했을걸세. $EK_a[R]$와 $E_R[M]$을(규약을 볼것) 도청한후에 가령 그것을 Z라고 하면 그는 계속 A처럼 가장할것이며 그 다음은 …》

제2편. 공개열쇠암호와 하쉬함수

제6장. 공개열쇠암호

공개열쇠암호의 개발은 암호학의 전 력사에서 가장 위대한 변혁이다. 암호학의 초창기부터 현 시대에 이르기까지 실제적으로 암호체계들은 초등적인 치환법이나 환자법에 기초하고 있다. 본질적으로 손계산에 의존하던 알고리듬의 리용이 수천년간 전통암호에서의 주요한 전진으로서 회전자식암호화/복호화기계가 개발되였다. 전자기계적인 회전자는 아주 복잡한 암호체계를 개발할수 있게 하였다. 콤퓨터의 출현에 의해 아주 복잡한 체계도 자료암호화표준(DES)으로 가장 높은 수준에서 계획되였다. 그러나 회전자기계와 DES가 많은 전진을 가져 왔지만 아직은 치환법이나 환자법의 기틀에 의존한것이였다.

공개열쇠암호는 종전의 모든것으로부터 벗어 나 근본적인 전변을 가져 왔다. 어떤 대상에 대한 공개열쇠알고리듬은 환자법이나 치환법과는 달리 수학적함수에 기초하고 있다. 보다 중요하게는 공개열쇠암호는 오직 하나의 열쇠만을 리용하는 대칭암호와는 달리 두개의 개별적열쇠들을 리용하는 비대칭암호라는것이다. 비대칭열쇠의 리용은 후에 보게 되겠지만 기밀성, 열쇠배포 및 인증분야에서 심오한 결과들을 가져 왔다.

본론에 들어 가기전에 공개열쇠암호와 관련되는 몇가지 공통적인 잘못된 견해들을 강조한다. 그 한가지는 공개열쇠암호가 전통암호보다 암호분석의 관점에서 더 안전하다는것이다. 이런 주장은 례하면 참고문헌[GARD77]에서 언급되였다. 사실 암호방식의 보안은 열쇠의 길이와 암호문을 분석하는데 드는 계산작업에 관계된다. 원리적으로 볼 때 암호분석을 저지시키는데서 전통암호와 공개열쇠암호중에서 어느것이 더 우월하다고 볼수 없다.

다음 잘못된 견해는 공개열쇠암호가 전통암호를 낡아 지게 하는 보편적기술이라는것이다. 이와는 반대로 현재 공개열쇠암호방식의 계산량적부하때문에 전통암호를 그만두게 될것이라는 예측은 없어 보인다. 공개열쇠암호창시자의 한사람이 내놓은바와 같이(참고문헌[DIFF88])《공개열쇠암호의 열쇠관리와 서명에서의 제한성은 거의 보편적으로 인정되고 있다》.

마지막으로 전통암호를 리용할 때의 열쇠배포쎈터와의 시끄러운 접촉을 공개열쇠암호를 리용할 때에는 열쇠배포가 간단해 졌다고 하는 생각이다. 사실은 일반적으로 중개대리인을 포함할것을 요구하며 포함된 절차는 전통암호에서 요구되는것보다 간단하거나 더 효과적인것이 못되는 규약들이 필요하다(참고문헌[NEED78]의 분석을 보시오).

이 장에서는 공개열쇠암호를 개괄한다. 우선 그의 개념적구조를 본다. 흥미 있는것은 이 기술들을 채용하여 실천적이라는것을 보여 주기도전에 그것이 개발되여 발표된것이다. 다음으로 공개열쇠암호가 가능하다는것을 보여 주는 가장 중요한 암호화/복호화알고리듬인 RSA알고리듬을 설명한다. 그다음에는 디피-헬만(Diffie-Hellman)의 열쇠교환에 대한 론의를 포괄한 공개열쇠체계의 열쇠관리 및 배송을 설명한다. 마지막으로 타원곡선암호에 대한 초보를 취급한다.

공개열쇠암호체계에서 대부분의 리론은 수론에 기초하고 있다. 이 장에서 취급하는 내용들을 리해하는데 반드시 수론의 지식이 필요한것은 아니다. 그러나 공개열쇠알고리듬에 대한 보다 완벽한 습득을 위해서는 수론의 지식이 요구된다. 7장에서는 이것을 개괄한다.

6.1 공개열쇠암호체계의 원리

공개열쇠암호는 전통암호에서 제기되는 가장 어려운 문제들중 2가지를 해결하기 위한 시도로부터 발족하였다. 첫번째 문제는 열쇠배포인데 5장에서 구체적으로 설명하였다.

거기에서 본바와 같이 전통암호에 의한 열쇠배포는 두 통신자가 (1) 자기들에게 배송될 열쇠를 이미 공유하거나 (2) 열쇠배포쎈터를 리용할것을 요구한다. 공개열쇠암호창시자의 한 사람인 디피(Diffie)는 두번째 요구가 암호학의 본질 즉 자기자신의 통신에서 총적안전성을 보장하는 능력을 부정한다고 생각해 내였다. 디피는 문헌[DIFF88]에서 다음과 같이 서술하였다.《암호체계의 사용자들이 자기의 열쇠를 할수 없이 공유하게 되여 침입자에 의해 손상당할바에야 손상된 KDC를 공유하는데 집중하였다면 신성불가침인 암호체계를 개발하는것이 무엇에 필요한 일인가?》

디피가 제기한 두번째 문제는 열쇠배포와는 관계 없는 《수자서명》이다. 암호를 군사분야에서만이 아니라 상업적 및 개인적목적으로 널리 리용한다면 전자편지와 문서들에도 종이문서에서 리용되는것과 같은 수표(서명)가 필요하게 된다. 즉 수자서명이 특정한 사람에 의해 보내졌다는것을 모든 사람들이 알수 있도록 규정하는 방법을 만들수 있는가 이것은 인증보다 좀 더 넓은 요구이며 그 특징과 차이를 10장에서 설명한다.

디피-헬만이 1976년에 수천년동안 내려 오던 암호학에 관한 선행한 모든 방식과는 근본적으로 다르며 우의 두 문제에 답을 주는 방법을 제기함으로써 돌파구를 열어 놓았다.

다음의 소제목에서는 공개열쇠암호의 전면적인 구조를 본다. 그다음에 이 방식의 중심인 암호화/복호화알고리듬에 대한 요구를 설명한다.

공개열쇠암호체계

공개열쇠알고리듬은 하나의 암호화열쇠와 그와는 다르지만 관련되는 복호화열쇠에 의거한다. 이 알고리듬은 다음과 같은 중요한 특성을 가진다.

- 암호알고리듬의 지식과 암호화열쇠만으로는 복호화열쇠를 구하는것이 계산량적으로 불가능하다.

그외에도 RSA와 같은 일부 알고리듬은 또한 다음과 같은 특성이 있다.

- 2개의 관련되는 열쇠중 임의의 하나를 암호화에, 다른것을 복호화에 리용할수 있다.

그림 6-1의 ㄱ(그림 2-1)은 공개열쇠암호화과정을 보여 준다. 기본단계는 다음과 같다.

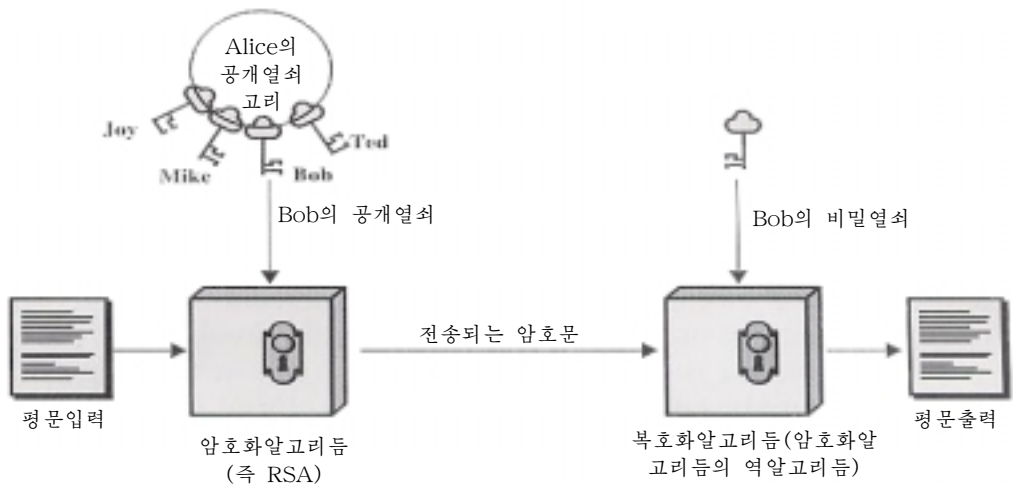
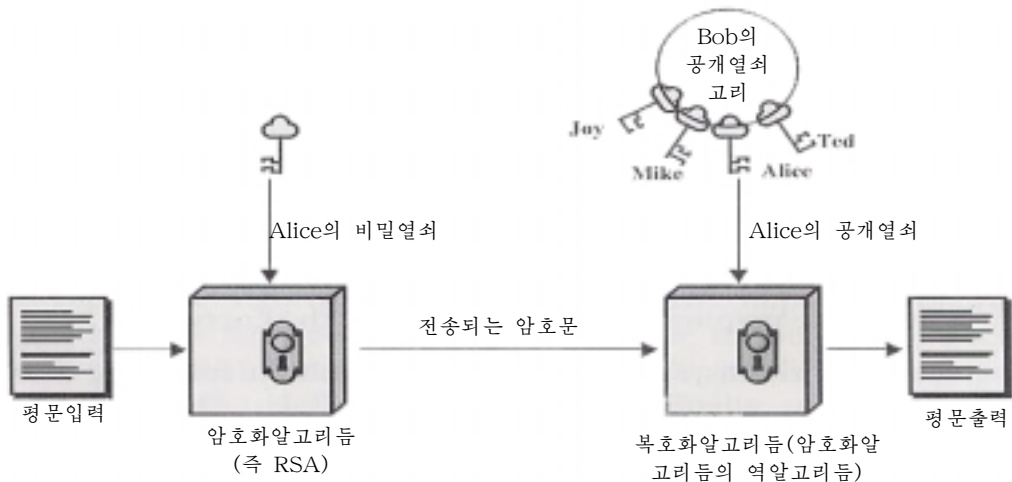
1. 망의 매 말단체계는 통보문의 암호화와 복호화에 쓰일 열쇠쌍을 생성한다.
2. 매 체계는 자기의 암호화열쇠를 공개등록기나 파일에 등록함으로써 그것을 공개한다. 이것이 공개열쇠이다. 다른 하나의 열쇠는 비밀로 간수한다.
3. A가 B에게 통보문을 보내려고 한다면 B의 공개열쇠를 리용하여 통보문을 암호화한다.
4. B가 통보문을 받았을 때 그것을 자기의 비밀열쇠를 리용하여 복호화한다. 오직 B만이 자기의 비밀열쇠를 알므로 다른 수신자는 통보문을 복호화할수 없다.

ㄱ) 암호

ㄴ) 인증

그림 6-1. 공개열쇠암호

이 방식에서 모든 가입자들은 공개열쇠를 참조하며 비밀열쇠는 매 가입자에 따라 국부적으로 생성되므로 결코 배송할 필요는 없다. 체계가 자기의 비밀열쇠를 소유하고 있는 동안 자신과 진행하는 통신은 안전하다. 임의의 시각에 체계는 자기의 비밀열쇠를 변경하여 그에 따르는 공개열쇠를 공개함으로써 낡은 공개열쇠와 교체한다.

표 6-1에는 전통암호와 공개열쇠암호의 중요한 내용의 일부를 제시하였다. 둘사이를 식별하기 위하여 일반적으로 전통암호에서 리용된 열쇠를 **전통비밀열쇠** 간단히 **비밀열쇠**라고도 하며 공개열쇠암호에서 리용된 두 열쇠를 각각 **공개열쇠, 비밀열쇠**라고 한다(다음의 표식들을 이 책전반에 걸쳐 모순없이 리용한다. 전통비밀열쇠를 K_m으로 표시한다. 여기서 m은 어떤 변경자이다. 가령 K_S는 대화열쇠이다. 사용자 A의 공개열쇠를 KU_a로, 대응하는 비밀열쇠를 KR_a로 표시한다. 평문 P의 암호화를 각각 $E_{K_m}[P]$, $E_{KU_a}[P]$ 및 $E_{KR_a}[P]$로 표시한 전통비밀열쇠, 공개열쇠 및 비밀열쇠로 진행할수 있다. 류사하게 암호문 C의 복호화를 각각 $D_{K_m}[C]$, $D_{KU_a}[C]$ 및 $D_{KR_a}[C]$로 표시한 전통비밀열쇠, 공개열쇠 및 비밀열쇠로 진행할수 있다).

표 6-1. 전통암호와 공개열쇠암호

전통암호	공개열쇠암호
다음의 작업이 요구된다. 1. 암호화와 복호화에 같은 알고리듬과 같은 열쇠가 리용된다. 2. 송수신자는 알고리듬과 열쇠를 공유한다.	다음의 작업이 요구된다. 1. 하나의 열쇠쌍 즉 암호화열쇠와 복호화열쇠로 암호화와 복호화를 위해 하나의 알고리듬이 리용된다. 2. 송수신자는 관련된 열쇠쌍(같지 않은) 중의 하나를 가지고 있어야 한다.
보안을 위해 요구되는것은 다음과 같다. 1. 열쇠를 비밀로 간수해야 한다. 2. 다른 정보를 리용하지 않는 한 통보문을 분석하는것은 불가능하거나 비현실적이여야 한다. 3. 알고리듬의 지식+암호문의 견본만으로는 열쇠를 구하는것이 불가능해야 한다.	보안을 위해 요구되는것은 다음과 같다. 1. 두 열쇠중의 하나는 비밀로 간수되여야 한다. 2. 다른 정보를 리용하지 않는 한 통보문을 분석하는것은 불가능하거나 비현실적이여야 한다. 3. 알고리듬의 지식+하나의 열쇠+암호문의 견본만으로는 다른 열쇠를 구하는것이 불가능해야 한다.

그림 6-2(그림 2-2와 비교하면서)를 리용하여 공개열쇠암호방식의 본질적요소들을 좀더 구체적으로 고찰하자. 평문통보문 $X=[X_1, X_2, \cdots\cdots, X_M]$을 내보내는 어떤 통보문 원천지 A가 있다. X의 M개 요소들은 어떤 유한자모의 문자들이다. 통보문은 목적지 B에로 전달된다. B는 관계되는 열쇠쌍 즉 공개열쇠 KU_b와 비밀열쇠 KR_b를 생성한다. KR_b는 B만이 알고 있지만 KU_b는 공개적으로 알수 있으므로 A가 참조할수 있다.

입구가 통보문 X와 암호화열쇠 KU_b일 때 A는 암호문 $Y=[Y_1, Y_2, \cdots\cdots, Y_N]$을 다음과 같이 만든다.

$$Y=E_{KU_b}(X)$$

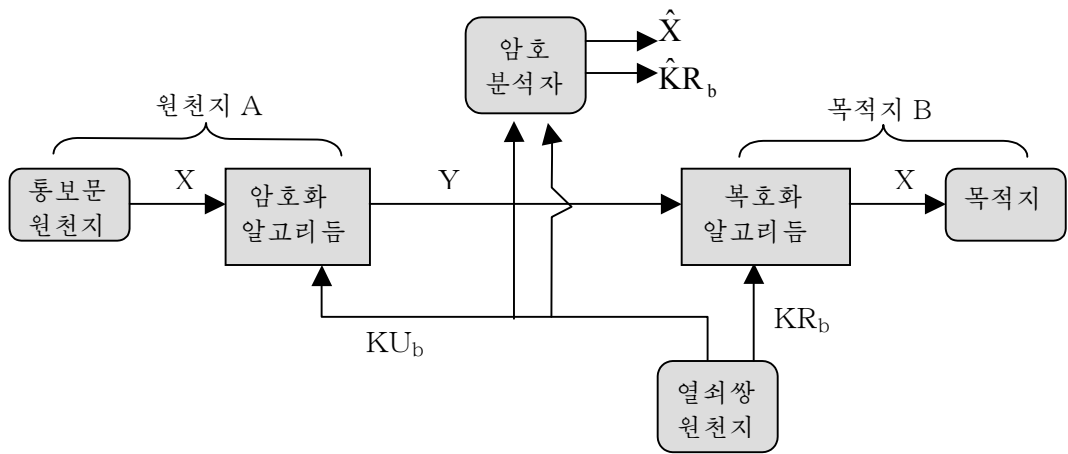



그림 6-2. 공개열쇠암호체계(안전성)

대응하는 비밀열쇠를 소유하고 있는 수신자는 다음과 같은 반대변환을 한다.

$$X=D_{KU_b}(Y)$$

Y를 관측하고 KU_b를 참조하지만 KR_b나 X를 참조하지 못하는 적은 X 또는 KR_b 혹은 둘 다 얻으려고 시도해야 한다. 적은 암호화알고리듬(E)과 복호화알고리듬(D)의 지식이 있다고 가정한다. 적이 오직 이 특별한 통보문에만 관심을 가진다면 노리는 초점은 평문근사 $\hat{X}$을 생성함으로써 X를 얻으려는데 둔다. 그러나 때때로 적은 근사 $\hat{KR}_b$를 생성하여 KR_b를 얻으려고 하는 경우에는 물론 미래의 통보문을 읽는데 관심이 있을것이다.

2개의 관련된 열쇠중 어느 하나는 암호화에, 다른것은 복호화에 리용된다는것을 이미 강조하였다. 이것은 오히려 각이한 암호방식을 실현할수 있게 한다. 그림 6-2에서 보여 준 방식이 기밀성이 된다면 그림 6-1의 ㄴ과 그림 6-3은 다음과 같이 공개열쇠암호를 리용하여 인증하는것을 보여 준다.

$$Y=E_{KR_a}(X)$$

$$X=D_{KU_a}(Y)$$

이 경우에 A는 B에게 통보문을 전송하기전에 A의 비밀열쇠를 리용하여 그것을 암호화한다. B는 A의 공개열쇠를 리용하여 통보문을 복호화한다. A의 비밀열쇠를 리용하여 통보문을 암호화하였으므로 오직 A만이 통보문을 준비하였다. 그러므로 완전히 암호화된 통보문이 **수자서명**으로서 종사한다. 그외에 A의 비밀열쇠를 참조함이 없이 통보문을 변경하는것은 불가능하다. 그래서 통보문은 원천지 그리고 자료완정성에 의하여 인증된다.

선행한 방식에서는 주제와 내용을 둘 다 타당하게 한다고 할지라도 아주 많은 기억기를 요구하는 통보문을 완전히 암호화해야 한다. 매 문서는 실천적목적을 위해 쓰이는 평문속에 간수되여야 한다. 또한 복사는 암호문속에 기억되며 원본과 내용을 대조하여 검증할수 있어야 한다. 같은 결과를 얻기 위한 보다 효과적인 방법은 문서들의 함수인 좀더 작은 블로크들을 암호화하는것이다. **인증자**라고 하는 이런 블로크는 인증자를 변경함이 없이 문서를 변경하는것은 불가능하다는 속성을 가져야 한다. 인증자를 송신자의 비밀열쇠로 암호화하면 그것은 원본,내용 그리고 렬을 검증하는 서명으로서 종사한다. 10장에서는 이 기술을 구체적으로 설명한다.

여기서 서술한 암호화처리가 기밀성이 없다는것을 강조하게 된다. 즉 보내는 통보문은 변경으로부터는 안전하지만 도청으로부터는 그렇치 못하다. 이것은 통보문의 나머지가 명백하게 전송되므로 통보문의 일부분에 기초한 서명인 경우에 명백하다. 지어 그림 6-3에서 보여 준바와 같이 완전한 암호화의 경우에는 임의의 관측자가 송신자의 공개열쇠를 리용하여 통보문을 복호화할수 있으므로 기밀성이 없다.

그러나 다음과 같이 공개열쇠방식의 중복리용에 의하여 인증기능과 기밀성을 둘 다 할수 있다.(그림 6-4)

$$Z=E_{KU_b}[E_{KR_a}(X)]$$
$$X=D_{KU_a}[D_{KR_b}(Z)]$$

이 경우에는 송신자의 비밀열쇠를 리용하여 통보문을 암호화하는것으로부터 시작한다. 이것은 수자서명을 제공한다. 다음으로 수신자의 공개열쇠를 리용하여 다시 암호화한다. 최종적인 암호문을 오직 관계되는 수신자만이 복호화할수 있는데 그에게만 대응하는 비밀열쇠가 있다. 이리하여 비밀이 담보된다. 이 방식의 불합리성은 공개열쇠알고리듬이 복잡하며 매 통신에서 두번이 아니라 네번 동작시켜야 하는것이다.

공개열쇠암호체계의 응용

설명하기전에 혼동을 피하기 위하여 공개열쇠암호의 한가지 내용을 명백히 해야 할 필요가 있다. 공개열쇠체계는 두 열쇠 즉 비밀열쇠와 공개열쇠를 가진 암호론적형태의 알고리듬의 리용에 의해 특징 지어 진다. 응용과 관련하여 송신자는 자기의 비밀열쇠 또는 수신자의 공개열쇠를 리용하거나 일정한 형태의 암호론적기능을 수행하기 위하여 둘 다 리용한다. 넓은 의미에서 공개열쇠암호체계의 리용을 다음과 같이 세가지로 분류할수 있다.

- **암호화/복호화:** 송신자는 수신자의 공개열쇠로 통보문을 암호화한다.
- **수자서명:** 송신자는 자기의 비밀열쇠로 통보문에 《서명》한다. 서명은 통보문에 또는 통보문에 관한 함수인 작은 자료블로크에 적용된 암호론적알고리듬에 의해 달성된다.
- **열쇠교환:**두 측은 협동하여 대화 또는 접촉열쇠를 교환한다. 하나 또는 둘 다의 비밀열쇠를 포함하는 여러가지 방식들이 가능하다.

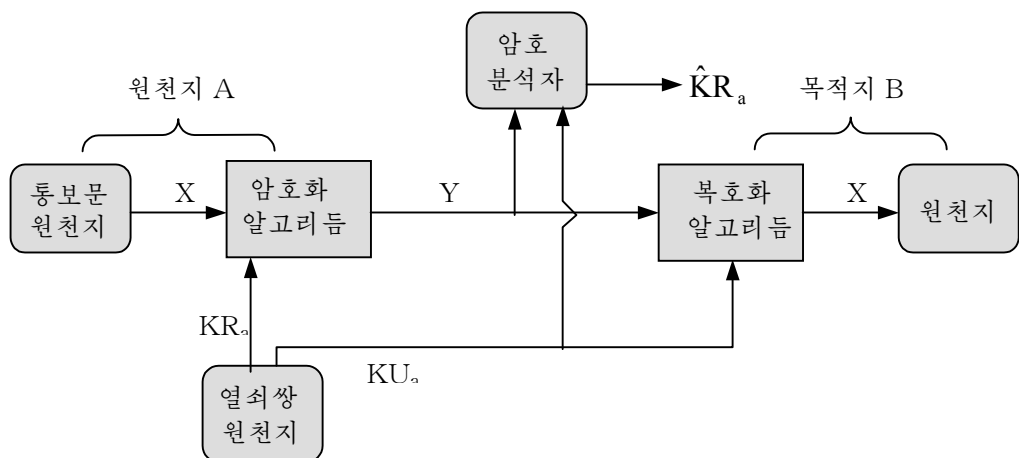



그림 6-3. 공개열쇠암호체계:인증

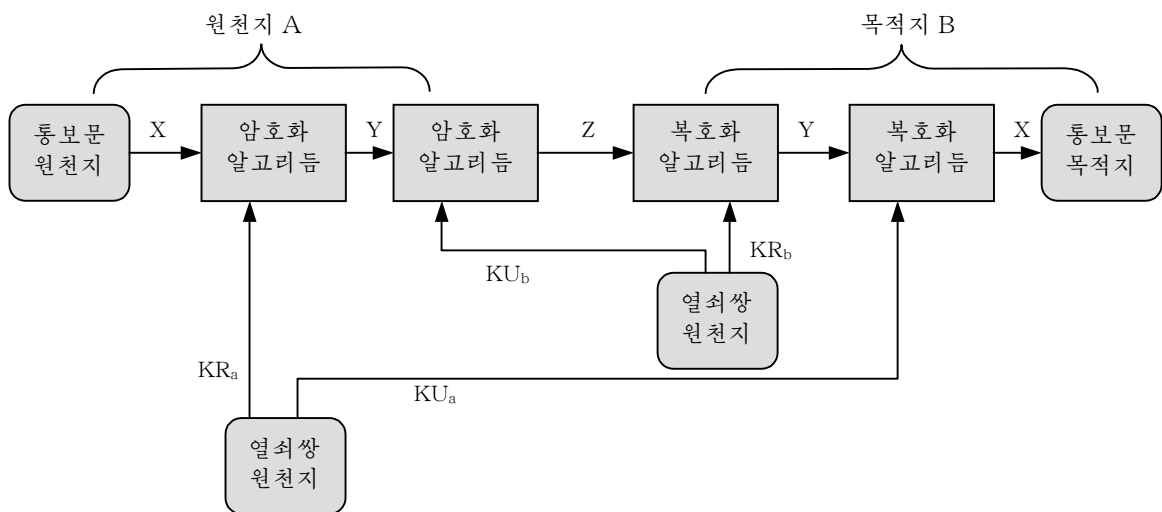



그림 6-4. 공개열쇠암호체계:안전성과 인증

어떤 알고리듬은 3개의 응용에 다 적합하지만 다른것들은 이 응용들중 둘 또는 하나에만 리용될수 있다. 표 6-2는 이 책에서 론의된 알고리듬이 지원하는 응용들을 보여 준다.

공개열쇠암호의 요구

그림 6-2로부터 6-4까지에서 보여 준 암호체계는 두개의 관련된 열쇠들에 기초한 암호론적알고리듬에 의존하고 있다. 디피와 헬만은 이런 알고리듬이 존재함을 밝히지 않고 이 체계를 주장하였다. 그러나 그들은 이런 알고리듬이 완전하게 되여야 한다는 다음과

같은 조건을 내놓았다[DIFF76b].

표 6-2. 공개열쇠암호체계의 응용

알고리듬	암호화/복호화	수자서명	열쇠교환
RSA	예	예	예
디피-헬만	아니	아니	예
DSS	아니	예	아니

1. B측이 쌍(공개열쇠 KU_b, 비밀열쇠 KR_b)을 생성하는것은 계산량적으로 간단하다.
2. 공개열쇠와 암호화할 통보문 M을 아는 송신자 A가 대응하는 암호문

$$C=E_{KU_b}(M)$$

을 생성하는것은 계산량적으로 간단하다.

3. 수신자 B가 원래의 통보문을 얻기 위한 비밀열쇠를 리용하여 다음과 같이 암호문을 복호화하는것은 계산량적으로 간단하다.

$$M=D_{KR_b}(C)=D_{KR_b}[E_{KU_b}(M)]$$

4. 공개열쇠 KU_b를 아는 적이 비밀열쇠 KR_b를 구하는것은 계산량적으로 불가능하다.
5. 공개열쇠 KU_b와 암호문 C를 아는 적이 원래의 통보문 M을 얻는것은 계산량적으로 불가능하다.

비록 유용하다 할지라도 모든 공개열쇠응용에는 필요하지 않은 6번째 요구를 다음과 같이 보충할수 있다.

6. 암호화와 복호화함수를 다음과 같이 임의의 순서로 적용할수 있다. 즉

$$M=E_{KU_b}[D_{KR_b}(M)]$$

공개열쇠암호의 개념이 제안된 때로부터 몇십년이 되도록 오직 이런 알고리듬하나만이 널리 접수되였다는 사실로부터 알수 있는바와 같이 이것은 간단치 않은 요구이다.

왜 이런 요구가 그렇게 간단치 않은가를 설명하기전에 우선 그것을 다시 고찰하자. 이 요구들은 한방향락호함수를 필요로 한다. 한방향함수는 1:1함수이고 정방향계산은 쉽지만 역방향계산은 불가능하다(한방향하쉬함수와 혼돈하지 말아야 한다. 한방향하쉬함수는 인증에 리용되는데(8장) 그것은 임의의 자료부분을 인수로 취하여 어떤 고정된 출구에로 넘긴다). 즉

$$Y=f(X) \quad \text{쉽다}$$
$$X=f^{-1}(Y) \quad \text{불가능하다}$$

일반적으로 쉽다는것은 입력길이에 관한 함수로서 다항식시간내에 풀수 있는 문제

를 의미한다. 이리하여 입력길이가 $n$bit이면 함수를 계산하는 시간은 n^a에 비례한다. 여기서 a는 고정된 상수이다. 이런 알고리듬을 클라스 P에 속한다고 한다. 용어 《불가능하다》는 대단히 모호한 개념이다. 일반적으로 어떤 문제를 풀기 위한 시간이 입력크기에 관한 함수로서 다항식시간보다 빨리 성장하면 그 문제는 불가능하다고 한다. 실례로 입력길이가 $n$bit이고 함수를 계산하는 시간이 2^n에 비례하면 그 문제는 불가능한것으로 고찰된다. 그런데 임의의 알고리듬이 이런 복잡성을 나타내는가를 결정하는것은 힘들다. 그래서 계산복잡성에 대한 전통적인 개념은 알고리듬의 최악의 경우 또는 평균경우의 복잡성이다. 이런 요구는 최악의 경우가 아니라 평균경우조차도 실제적으로 모든 입력에 대하여 반전하는것이 불가능한 암호학에서는 의의가 없다. 이런 개념들에 대한 몇가지 간단한 결과를 부록 6에 주었다.

이제는 한 방향으로 계산하는것은 쉽지만 어떤 보충적인 정보를 모른다면 다른 방향으로 계산하는것이 불가능한 한방향락호함수를 정의하자. 보충적인 정보를 가지고 역방향은 다항식시간내에 계산될수 있다. 다음과 같은것을 개괄할수 있다. 한방향락호함수는

k와 X가 알려 지면 $Y=f_k(X)$는 쉽다
k와 Y가 알려 지면 $X=f_k^{-1}(Y)$는 쉽다
Y는 알지만 k를 모르면 $X=f_k^{-1}(Y)$는 불가능

인 1:1함수 f_k들의 족이다. 이리하여 실천적인 공개열쇠방식의 개발은 적합한 한방향락호함수의 발견에 관계된다.

공개열쇠암호의 분석

전통암호와 마찬가지로 공개열쇠암호방식은 힘내기공격에 약하다. 대응책 역시 마찬가지이다. 즉 큰 열쇠를 리용한다. 그러나 공개열쇠암호에는 한가지 고려해야 할 난점이 있다. 공개열쇠체계는 일종의 가역인 수학적함수의 리용에 의존한다. 이런 함수의 계산복잡성은 열쇠의 비트수에 선형적으로가 아니라 그보다 빨리 성장한다. 이리하여 열쇠의 크기는 실천적으로 힘내기공격이 불가능하도록 충분히 커야 하지만 실천적인 암호화와 복호화를 위해서는 충분히 작아야 한다. 실천에서 제기되는 열쇠의 크기는 실천적으로 힘내기공격을 불가능하게 하지만 암호화/복호화의 결과는 일반목적으로 쓰는데는 너무 굼뜨게 나타난다. 그 대신에 앞에서 언급한바와 같이 공개열쇠암호를 현재 열쇠관리와 서명에 적용하는것이 합리적이다.

다른 한 형태의 공격은 공개열쇠가 주어 졌을 때 비밀열쇠를 계산하는 방법을 구하는것이다. 이런 형태의 공격이 특별한 공개열쇠알고리듬에 대하여 불가능하다는것은 수학적으로 증명되지 않았다. 이리하여 널리 쓰이는 RSA알고리듬을 포함하여 임의의 주어 진 알고리듬이 의문에 붙여 졌다. 암호분석의 력사는 하나의 관점으로는 풀수 없는것처럼 보이는 문제도 완전히 다른 각도에서 보면 풀이를 구할수 있다는것을 보여 준다.

마지막으로 공개열쇠체계에 독특한 한가지 형태의 공격이 있는데 본질에 있어서는 일정한 통보문공격이다. 실례로 56bitDES열쇠만으로 이루어 진 통보문을 보냈다고 가정하자. 적은 공개열쇠를 리용하여 가능한 열쇠모두를 암호화하고 전송된 암호문을 대조함으로써 임의의 통보문을 복호화할수 있다. 이리하여 공개열쇠방식에서 열쇠의 크기는 문제로 되지 않으며 공격은 56bit열쇠에 대한 힘내기공격에 귀착된다. 이런 통보문에 어떤 우연비트를 첨가함으로써 이 공격을 좌절시킬수 있다.

6.2 RSA알고리듬

디피와 헬만은 최초의 론문[DIFF76b]에서 암호학에 대한 새로운 방식을 받아 들이였는데 사실상 공개열쇠체계에 대한 요구를 만족시키는 암호알고리듬은 암호학자들의 도전에 부딪쳤다. 이 도전에 대한 첫 응답중의 하나가 MIT에서 리베스트, 샤미르, 아들레만(Rivest, Shamir, Adleman)에 의하여 1977년에 개발되였으며 1978년에 처음으로 출판되였다[RIV78]. RSA방식은 공개열쇠암호에 대한 일반목적방식을 유일하게 광범히 접수하고 실현하였으므로 가장 우수한것이다.

RSA방식은 어떤 n에 대하여 평문과 암호문이 0과 n-1사이의 옹근수인 블로크암호이다. 이 절에서는 알고리듬의 해석으로부터 시작하면서 좀 더 구체적으로 RSA를 설명한다. 다음으로 RSA의 계산량적 및 암호분석적따름을 설명한다.

알고리듬의 서술

리베스트, 샤미르, 아들레만이 개발한 이 방식은 제곱식을 리용한다. 평문은 블로크들로 암호화되며 매 블로크는 어떤 수 n보다 작은 2진값이다. 즉 블로크의 크기는 $\log_2(n)$을 넘지 않는다. 실천에서 블로크의 크기는 2^kbit이다. 여기서 $2^k < n \le 2^{k+1}$이다. 암호화와 복호화는 어떤 평문블로크 M과 암호문블로크 C에 대하여 다음과 같은 형태를 가진다.

$$C = M^e \bmod n$$
$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

두 송수신자는 n의 값을 알아야 한다. 송신자는 e의 값을 알며 오직 수신자만이 d의 값을 안다. 이리하여 이것은 공개열쇠 KU=$\{e,n\}$과 비밀열쇠 KR=$\{d,n\}$인 공개열쇠알고리듬이다. 공개열쇠암호를 만족시키는 이 알고리듬에 대하여 다음과 같은 요구들이 성립되여야 한다.

1. 모든 M < n에 대하여 $M^{ed} = M \bmod n$인 e, d, n의 값들을 구하는것이 가능하다.
2. M < n의 모든 값들에 대하여 M^e와 C^d 을 계산하는것이 상대적으로 쉽다.
3. e와 n이 주어 졌을 때 d를 구하는것은 불가능하다.

이제 첫 문제에 집중하고 후에 다른 문제들을 고찰하자.

$$M^{ed} = M \bmod n$$

형태의 식을 구하는것이 필요하다.

7장(식 7-7)에서 제기된 오일레르정리의 따름은 다음과 같은 성질을 담보한다. 즉 주어 진 두 씨수 p와 q 그리고 n=pq 및 0<m<n인 두 옹근수 n과 m 그리고 임의의 옹근수 k에 대하여

$$m^{k\phi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \bmod n$$

이다. 여기서 $\phi(n)$은 오일레르함수 즉 n보다 작으면서 n과 서로 소인 정의옹근수의 수이다. 7장에서 씨수 p, q에 대하여 $\phi(pq)=(p-1)(q-1)$임을 보여 준다. 이리하여

$$ed=k\phi(n)+1$$

이면 우리의 주장이 성립한다는것을 알수 있다. 이것은 다음의것과 동등하다.

$$ed \equiv 1 \mod \phi(n)$$
$$d \equiv e^{-1} \mod \phi(n)$$

즉 e와 d는 mod $\phi(n)$에 관하여 곱하기에 관한 역수이다. 모드산수의 규칙에 의하여 이것은 오직 d(따라서 e)가 $\phi(n)$과 서로 소일 때만 참이다. 동등하게 $\gcd(\phi(n), d)=1$이다.

이제는 RSA방식을 정식화하여 보자. 구성요소는 다음과 같다.

두 씨수 p, q	(비밀로 선택)
$n=pq$	(계산하여 공개)
$\gcd(\phi(n), e)=1$이고 $1<e<\phi(n)$인 e	(선택하여 공개)
$d \equiv e^{-1} \mod \phi(n)$	(비밀로 계산)

비밀열쇠는 $\{d, n\}$이며 공개열쇠는 $\{e, n\}$이다. 사용자 A가 자기의 공개열쇠를 공개하며 사용자 B는 통보문 M을 A에게 보내려고 한다고 가정하자. 이때 B는 $C = M^e \pmod n$을 계산하여 C를 전송한다. 이 암호문을 접수하면 사용자 A는 $M = C^d \pmod n$을 계산하여 복호화한다.

이 알고리듬의 타당성을 개괄하는것은 응당하다. 다음과 같이 e와 d를 선택한다.

$$d \equiv e^{-1} \mod \phi(n)$$

그러므로

$$ed \equiv 1 \mod \phi(n)$$

따라서 ed는 $k\phi(n)+1$의 형태이다. 그런데 7장에서 제기한 오일레르정리의 따름에 의하여 씨수 p와 q, 옹근수 $n = pq$와 $0<M<n$인 M이 주어 졌을 때

$$M^{k\phi(n)+1} = M^{k(p-1)(q-1)+1} \equiv M \mod n$$

그래서 $M^{ed} = M \mod n$이다. 이제는

$$C = M^e \mod n$$
$$M = C^d \mod n \equiv (M^e)^d \mod n \equiv M^{ed} \mod n \equiv M \mod n$$

그림 6-5는 RSA알고리듬을 개괄한다. 그림 6-6에서 실례를 보여 준다. 이 실례에 대한 열쇠를 다음과 같이 생성하였다.

1. 두 씨수 $p=7$과 $q=17$을 택한다.
2. $n=pq=7\times17=119$를 계산한다.
3. $\phi(n)=(p-1)(q-1)=96$을 계산한다.

4. $\phi(n)$=96과 서로 소이며 $\phi(n)$보다 작은 e를 택한다. 즉 e=5.
5. de=1 mod 96이고 d<96인 d를 구한다. 77×5=385=4×96+1이므로 정확한 값은 d=77이다.

열 쇠 생 성

p, q를 택한다.	p와 q는 둘 다 씨수이다.
$n=p\times q$를 계산한다.	
$\phi(n)=(p-1)(q-1)$을 계산한다.	
옹근수 e를 택한다.	$\gcd(\phi(n), e)=1;\ 1<e<\phi(n)$
d를 계산한다.	$d = e^{-1} \bmod \phi(n)$
공개열쇠	KU={e, n}
비밀열쇠	KR={d, n}

암 호 화

평문	$M < n$
암호문	$C = M^e \pmod{n}$

복 호 화

평문	C
암호문	$M = C^d \pmod{n}$

그림 6-5. RSA알고리듬

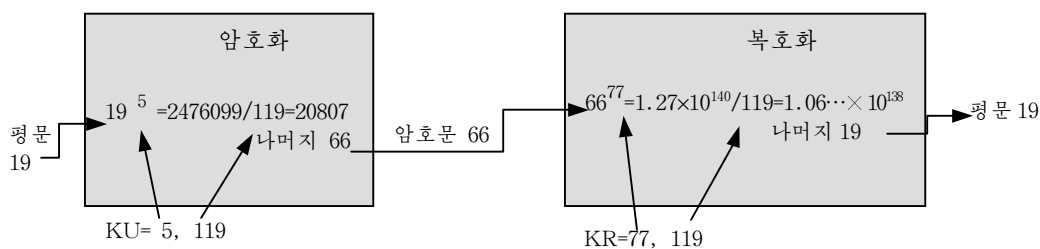



그림 6-6. RSA알고리듬의 실례

결과의 열쇠들은 공개열쇠 KU={5, 119}와 비밀열쇠KR={77, 119}이다. 이 실례는 평문입력 M=19에 이 열쇠들을 리용한것을 보여 준다. 암호화에 대하여 19를 5제곱하면 결과는 2476099로 된다. 119로 나누어 나머지를 구하면 66이다. 따라서 $19^5 \equiv 66 \bmod 119$이며 암호문은 66이다. 복호화에 대하여 $66^{77} \equiv 19 \bmod 119$를 구한다.

계산량적관점

이제 RSA를 리용하는데 요구되는 계산복잡성을 고찰하자. 실제적으로 열쇠생성과 암호화/복호화의 두가지가 제기된다. 우선 암호화와 복호화처리를 고찰하고 그다음에 열쇠생성을 고찰한다.

암호화와 복호화

RSA의 암호화와 복호화는 둘 다 모드 n에 관한 옹근수제곱으로 제기된다. 제곱지수가 모드 n에 관하여 옹근수우에서 진행되면 중간값이 담보될것이다. 다행히도 다음과 같은 모드산수의 성질을 리용할수 있다.

$$(a \bmod n)\times(b \bmod n) \bmod n=(a\times b) \bmod n$$

이리하여 mod n에 관한 중간결과를 얻을수 있다.

다른 하나는 RSA가 큰 제곱지수를 취급하므로 제곱연산의 효과성을 고찰하는것이다. 효과성이 어떻게 증가되는가를 보기 위하여 x^{16}을 계산하는것을 고찰하자. 간단한 방식은 다음과 같이 15번 곱하는것이다.

$$x^{16} = x\times x\times x\times x\times x\times x\times x\times x\times x\times x\times x\times x\times x\times x\times x\times x$$

그러나 련속적으로 x^2, x^4, x^8, x^{16}을 이르는 매 부분결과의 2제곱을 반복적으로 취하면 오직 4번의 곱하기만으로 같은 최종결과를 얻을수 있다.

보다 더 일반적으로 a와 정의옹근수 m에 대하여 값 a^m을 구하려고 한다고 가정하자. m을 2진수 $b_k b_{k-1}\ \cdots\ b_0$으로 표현하면 다음과 같다.

$$m = \sum_{b_i \neq 0} 2^i$$

```
c←0; d←0
for i←downto 0
   do c←2×c
         d←(d×d) mod n
         if b:=1
              then c←c+1
                 d←(d×a) mod n
return d
```

그림 6-7. a^b mod n을 계산하는 알고리듬

그러므로

$$a^m = a^{\left(\sum_{b_i\neq 0} 2^i\right)} = \prod_{b_i \neq 0} a^{\left(2^i\right)}$$

$$a^m \bmod n = \left[\prod_{b_i \neq 0} a^{(2^i)}\right] \bmod n = \prod_{b_i \neq 0}\left[a^{(2^i)} \bmod n\right]$$

따라서 그림 6-7에서 보여 준 $a^b \bmod n$을 계산하는 알고리듬을 개발할수 있다(이 알고리듬은 오랜 력사를 가진다. 이 특별한 준부호식은 [CORM90]에 있다).

그림 6-8은 이 알고리듬의 실행실례를 보여 준다. C는 필요 없다는것을 강조한다. 즉 그것은 보충적인 목적에 쓰인다. C의 최종값은 제곱지수의 값이다.

열쇠생성

공개열쇠암호체계를 적용하기전에 매 가입자는 열쇠쌍을 생성해야 한다. 이것은 다음의 과제들을 포함한다.

- 두 씨수 p와 q를 구하기
- e와 d중 하나를 택하고 다른것을 구하기

우선 p와 q의 선택을 고찰하자. $n=pq$의 값이 유력한 적에게 알려 지게 되므로 전면적인 방법으로부터 p와 q의 발견을 방지하기 위하여 이러한 씨수들은 충분히 큰 모임(즉 p와 q는 큰 수이여야 한다.)으로 선택되여야 한다. 다른 한편 큰 씨수를 구하는데 리용된 방법은 효과적이여야 한다.

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
d	7	49	157	526	160	241	298	166	67	1

그림 6-8. $a^b \bmod n$을 위한 고속모드제곱연산알고리듬의 결과
a=7, b=560=1000110000, n=561인 경우

현재 임의의 큰 씨수를 생성해 내는 적합한 기술이 없으므로 이 문제를 처리하는 어떤 다른 수단이 요구된다. 일반적으로 리용된 절차는 요구된 크기의 홀수를 우연적으로 취하여 그 수가 씨수인지 아닌지를 검사하는것이다. 그렇지 않으면 취한 수가 씨수로 판단될 때까지 란수를 계속 취한다.

씨수판정의 변종들이 개발되여 있다(즉 일련의 이런 변종으로서 KNUT98을 보시오). 대부분은 여전히 그 판정이 확률적이라는것이다. 즉 이 판정은 순전히 주어 진 옹근수가 확률적씨수임을 결정한다. 이렇게 확신성이 결핍되여 있음에도 불구하고 이 판정은 요구된것과 1.0정도에 가까운 확률로 실행될수 있다. 실례로 보다 효과적이고 극적인 알고리듬의 하나는 밀러-라빈(Miller-Rabin)알고리듬으로서 7장에서 설명한다. 이 알고리듬과 대부분의 이런 알고리듬들에 대하여 주어 진 옹근수 n이 씨수인가 아닌가를 판정하는 절차는 n과 어떤 우연적으로 선택된 옹근수 a를 포함하는 일련의 계산을 진행하는 것이다. n이 이 판정에 《실패》하면 n은 씨수가 아니다. n이 이 판정을 《통과》하면 n은 씨수일수도 있고 아닐수도 있다. a로서 우연적으로 선택된 값에 대하여 이런 판정들

을 여러번 통과하면 사실상 n이 씨수일 가능성이 높다.

개괄하면 씨수를 포착하는 절차는 다음과 같다.

1. 홀수 n을 우연적으로 취한다(즉 모조란수생성기를 리용하여).
2. 옹근수 $a<n$을 우연적으로 취한다.
3. 밀러-라빈과 같은 확률적씨수판정을 진행한다. n이 이 판정에 실패하면 n을 거부하고 단계 1로 이행한다.
4. n이 충분한 회수의 판정을 통과하였다면 n을 접수한다. 그렇지 않으면 단계 2에로 이행한다.

이것은 좀 지루하다. 그러나 이 처리는 새로운 쌍(KU, KR)이 요구될 때만 상대적으로 드물게 진행된다는것을 상기하시오.

씨수를 발견하기전에 얼마나 많은 수들이 거부되는가를 론의할 가치가 있다. 씨수정리로 알려 진 수론의 결과는 n에 가까운 씨수는 평균적으로 매 $\ln(N)$개 옹근수내에 놓여 있다. 이리하여 씨수를 발견하기전에 평균적으로 $\ln(N)$개 옹근수를 차례로 판정해야 한다. 실제적으로는 모든 짝수들이 즉시에 거부되므로 정확한것은 $\ln(N)/2$이다. 실례로 2^{200}크기의 씨수를 탐색하려고 한다면 대략 $\ln(2^{200})/2=70$개의 시행이 씨수를 발견하는데 리용될것이다.

씨수 p와 q를 결정하면 열쇠생성의 처리는 e의 값을 택하고 d를 계산 또는 반대로 d의 값을 택하고 e의 값을 계산하는것으로 완성된다. 전자를 가정한다면 $\gcd(\phi(n), e)=1$인 e를 택한 다음에 $d = e^{-1} \bmod \phi(n)$을 계산할것을 요구한다. 다행히도 동시에 두 수의 최대공통약수를 계산하는 하나의 알고리듬이 있어서 gcd가 1이면 한 옹근수를 모드로 하여 다른 한 옹근수의 역수를 구한다. 확장된 유클리드알고리듬이라고 하는 이 알고리듬은 7장에서 설명한다. 이리하여 이 절차는 $\phi(n)$과 서로 소인 수를 발견할 때까지 $\phi(n)$과 매개를 판정하는 란수렬을 생성하는것이다. 다시 다음의 질문을 할수 있다. 리용가능한 수 즉 $\phi(n)$과 서로 소인 수를 구하기 위하여 얼마나 많은 란수를 판정해야 하는가? 두 란수가 서로 소일 확률은 0.6정도임을 쉽게 알수 있다. 이와 같이 적합한 옹근수를 구하는데는 극히 적은 판정이 요구된다(문제 7-1을 보시오).

RSA의 보안

RSA알고리듬을 공격하는 다음과 같은 3개의 가능한 방식들이 있다.

- **힘내기공격:** 이것은 가능한 비밀열쇠모두를 시도하는것이다.
- **수학적공격:** 두 씨수의 적을 인수분해하는 품과 동등한 일련의 방식들이 있다.
- **시간공격:** 이것은 복호화알고리듬의 실행시간에 의존한다.

폭력방식에 대처한 방어는 다른 암호체계에서처럼 RSA에서도 같다. 즉 큰 열쇠공간을 리용한다. 이리하여 e와 d의 비트수가 크면 클수록 좋다. 그러나 열쇠생성과 암호화/복호화를 둘 다 포함한 계산은 복잡하므로 열쇠의 크기가 클수록 체계의 실행속도는 떠진다.

아래에서는 수학적공격과 시간공격을 개괄한다.

인수분해문제

RSA를 수학적으로 공격하는 방식을 다음과 같이 3개로 분류할수 있다.

• 인수 n을 2개의 씨인수로 분해한다. 이것은 $d = e^{-1} \pmod{\phi(n)}$을 구할수 있게 하는 $\phi(n) = (p-1)\times(q-1)$을 계산할수 있다.
• p와 q를 결정함이 없이 직접 $\phi(n)$을 구한다. 이것은 다시 $d = e^{-1} \pmod{\phi(n)}$을 구할수 있다.
• $\phi(n)$을 구함이 없이 직접 d를 구한다.

대부분의 RSA암호분석자들의 론의는 n을 두 씨인자로 분해하는 과제에 집중하였다. 주어 진 n에 대하여 $\phi(n)$을 구하는것은 n을 씨인수분해하는것과 동등하다[RIBE96]. 현재 알려 진 알고리듬에 대하여 주어 진 e와 n으로부터 d를 구하는것은 적어도 인수분해문제와 같은 시간소비로 나타난다[KALI95]. 따라서 인수분해성능을 RSA의 보안을 평가하는 기준척도로 리용할수 있다.

큰 씨인수들을 가지는 큰 n에 대한 인수분해는 힘든 문제이지만 그것을 리용하는것만큼은 힘들지 않다. 이에 대한 파피실례는 다음과 같다. 1977년에 3명의 RSA창시자들은 암호문을 분석하는 독자들에게 도전하였다. 그들은 평문의 복호에 100$의 상금을 제공하기로 하고 이것을 4천만년내에는 풀지 못할것으로 예언하였다. 1994년 4월 인터네트상에서 작업하는 팀이 단지 8개월동안 작업한 다음에 상금을 청구하였다. 이 도전에는 129자리의 10진수 즉 대략 428bit크기의 공개열쇠(n의 길이)를 리용하였다. 그동안에 그들은 바로 DES로 하였으며 RSA집단은 100, 110, 120과 같은 자리수의 열쇠크기를 가진 암호문에 도전하였다. 최후에 만난 도전은 130자리의 열쇠길이를 가진 RSA-130도전이다[COWE96]. 표 6-3은 날자별 결과를 보여 준다. 효과수준은 MIPS년으로 측정되여 있다. 즉 초당 100만개 처리기를 한해에 실행하는것으로서 대략 3×10^{13}개를 실행한다. 200MHz의 Pentium이 대략 50-MIPS이다.

표 6-3에서 파피인자는 리용방법에 관계된다. 최근까지만 하여도 인수분해공격은 2차채법이라고 하는 방식을 리용하였다. RSA-130에 대한 공격은 보다 새로운 알고리듬 즉 일반화된 수채채법(GNFS)을 리용하였으며 다만 10%의 계산효과로 RSA-129보다 큰 수를 인수분해할수 있었다.

표 6-3. 인수분해과정

십진자리수	대략비트수	달성날자	MIPS년	알고리듬
100	332	1991.4	7	2차채법
110	365	1992.4	75	2차채법
120	398	1993.6	830	2차채법
129	428	1994.4	5000	2차채법
130	431	1996.4	500	일반화된수체채법

열쇠크기가 커질수록 두가지 문제 즉 계산능력의 증가와 인수분해알고리듬의 개선이 제기된다. 여러가지 알고리듬에로의 이동은 매우 높은 속도로 결과를 나타낸다는것을 보았다. GNFS로 보다 새로운 개선을 기대할수 있으며 따라서 좋은 알고리듬을 리용할수

있다. 사실상 관련된 알고리듬 즉 특수한 수채채법(SNFS)은 일반화된 수채채법보다 상당히 빨리 특수한 형태의 수를 인수분해할수 있다. 그림 6-9에서는 두 알고리듬의 성능을 비교하였다. 일반적인 인수분해성능이 대략 SNFS와 같다는 사실을 기대할수 없게 되였다[ODLY95]. 이리하여 RSA의 열쇠크기를 택하는데 주의를 돌려야 한다. 가까운 앞날에 1024～2048bit까지의 령역에서 열쇠크기가 합리적일것으로 추측된다.

n의 크기를 설명하는것외에 일련의 다른 제한들이 제기되였다. 보다 쉽게 인수분해될수 있는 n의 값을 피하기 위하여 알고리듬의 제창자들은 p와 q에 대하여 다음과 같은 제한을 제기하였다.

1. p와 q는 오직 몇개 수자들만이 차이난다. 이리하여 p와 q는 둘 다 10^{75}부터 10^{100}까지의 정도이다.
2. $(p-1)$과 $(q-1)$은 둘 다 큰 씨인수를 가진다.
3. $\gcd(p-1, q-1)$은 작다.

이외에도 $e<n$이고 $d<n^{1/4}$이면 d를 쉽게 구할수 있다[WIEN90].

시간공격

암호학적알고리듬의 보안을 론하는것이 얼마나 어려운가에 대한 다른 한가지 고찰이 요구된다면 그것은 시간공격의 출현이다. 파울 코커(Paul Kocher)는 콤퓨터가 통보문을 분석하는데 얼마나 오랜 시간을 소비하는가를 측정하여 비밀열쇠를 구하는것을 보여 주었다[KOCH96]. 시간공격은 RSA에는 적용할수 없으나 다른 공개열쇠암호체계에 적용할수 있다. 이 공격은 두가지 리유로 하여 경보를 울린다. 그것은 임의의 방향에서 들어오며 암호문전용공격이기때문이다.

시간공격은 어떤 사람이 수자에서 수자에로 다이얄을 얼마나 오래동안 돌리는가를 관측함으로써 안전성의 조합을 알아 맞추는 도적과 류사하다. 그림 6-7의 모드제곱연산알고리듬을 리용하여 이 공격을 설명할수 있다. 그러나 이 공격은 고정된 시간내에는 실행하지 못하는 임의의 실현으로 작업하는것을 갱신할수 있다. 이 알고리듬에서 모드제곱연산은 매 반복에서 실행된 하나의 곱하기와 매 1bit에 대하여 실행된 보충적인 곱하기에 의하여 비트별로 달성된다.

코커가 자기론문에서 이 공격은 극단한 경우에 리해하는것이 가장 단순하다는것을 강조하였다. 목표체계가 거의 모든 경우에는 아주 빠르지만 일부 경우에는 완전한 평균 모드제곱연산보다 더 많은 시간을 취하는 모드곱하기함수를 리용한다고 가정하자. 이 공격은 제일 왼쪽비트 b_k부터 시작하여 비트별로 진행된다. 첫 j개 비트를 안다고 가정하자(완전한 제곱지수를 구하기 위하여 j=0부터 시작하여 완전한 제곱지수를 알 때까지 공격을 반복한다). 주어 진 암호문에 대하여 공격자는 for순환의 첫 j개 반복을 완성할수 있다. 그다음단계의 조작은 미지의 통보문비트에 의존한다. 비트가 설정되면 $d \leftarrow (d \times a) \bmod n$을 실행할것이다. a와 d의 값에 대하여 모드곱하기는 매우 느리며 공격자는 그것이 어느것인가를 안다. 그러므로 이 특별한 반복이 1bit로 늦어 질 때 복호화알고리듬을 실행하는데 관측된 시간이 항상 느리면 이 비트는 1이라고 가정한다. 완전한 알고리듬에 대하여 일련의 관측된 시간이 빠르면 이 비트는 0이라고 가정한다.

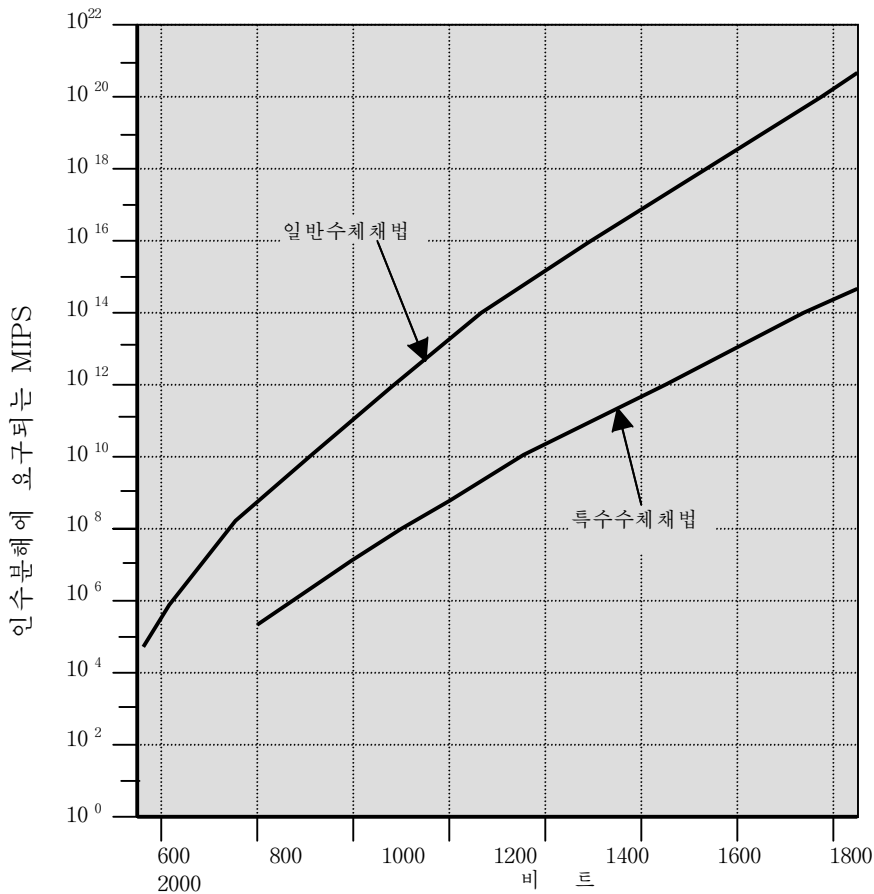



그림 6-9. 인수분해에 요구되는 MIPS년

실천에서는 모드제곱연산의 실현이 어떤 극적인 시간변화를 가지지 않는데 한 반복의 실행시간은 완전한 알고리듬의 평균실행시간을 릉가할수 있다. 그럼에도 불구하고 이 공격이 현실적인것으로 되도록 하는 충분한 변종이 있다. 구체적인것은 [KOCH96]을 보시오.

시간맞추기공격이 심각한 위협이라고 할지라도 다음과 같은것을 포괄하여 리용할수 있는 단순한 대응책이 있다.

- **불변지수시간:** 모든 제곱연산들은 결과를 귀환하기전에 같은 량의 시간을 취한다. 이것은 단순한 고정이지만 성능을 저하시킨다.
- **불규칙지연:** 보다 좋은 성능은 시간맞추기공격을 혼동시키는 제곱연산알고리듬에 불규칙지연을 첨가함으로써 달성될수 있다. 코커는 방어자가 충분한 잡음을 제거하지 않으면 공격자는 여전히 불규칙지연을 보상하는 보충적인 측정을 수집함으로써 성공한다는것을 강조하였다.

- **현혹시키기:** 제곱연산을 진행하기전에 란수로 암호문을 곱한다. 이 처리는 어느 암호문비트가 콤퓨터내부에서 처리되는가를 알아 내려는데로부터 공격자를 방지하므로 시간맞추기공격에서 본질적인 비트별분석을 방지한다.

RSA자료보안은 현혹적인 기능들을 그의 적(곱하기)들로 병합한다. 공개열쇠연산 M=C^d mod n은 다음과 같이 실현된다.

1. 0과 $n-1$사이의 비밀란수 r를 생성한다.
2. C' = $C r^e$ mod n을 계산한다. 여기서 e는 공개제곱지수이다.
3. 보통의 RSA실현으로 M'= $(C')^d$ mod n을 계산한다.
4. M =M' r^{-1} mod n을 계산한다. 이 식에서 r^{-1}은 mod n에 관한 r의 곱하기역수이다. 이 개념에 대한 론의는 7장을 보시오. 이것은 r^{ed} mod n = r mod n을 관측함으로써 정확한 결과라는것을 보여 줄수 있다.

RSA자료보안은 현혹적인것에 대하여 2～10%의 성능벌칙을 기록한다.

6.3 열쇠관리

5장에서 비밀열쇠의 배송문제를 설명하였다. 공개열쇠암호의 중요역할의 하나는 열쇠배포문제를 해결하는것이다. 여기에는 실제적으로 다음과 같은 공개열쇠암호의 리용에 관한 두가지 각이한 관점이 있다.

- 공개열쇠의 배송
- 공개열쇠암호를 비밀열쇠배포에 리용

이것을 실제로 설명하여 보자.

공개열쇠배포

공개열쇠배포를 위한 일련의 기술들이 제기되였다. 실제적으로 이런 제의모두를 다음과 같은 일반방식으로 종합할수 있다.

- 공개적인 발표
- 공개적으로 리용할수 있는 목록
- 공개열쇠 책임자
- 공개열쇠확인

공개열쇠의 공개적인 발표

이 측면에서 공개열쇠암호의 특징은 공개열쇠가 공개적이라는것이다. 이리하여 RSA와 같이 어떤 방대하게 접수된 공개열쇠알고리듬이 있다면 임의의 가입자는 자기의 공개열쇠를 임의의 다른 가입자에게 보내거나 그것을 대중에게 상세하게 발표할수 있다(그림 6-10). 실례로 RSA를 리용하는 PGP(12장에서 론의한다.)의 급격한 류행때문에 많은 PGP사용자들은 USENET직결토론회나 인터네트우편목록과 같은 공개토론회에 보

내는 통보문에 자기의 공개열쇠를 첨가하는 방식을 리용하였다.

이 방식이 편리하다고 할지라도 중요한 약점을 가지고 있다. 즉 임의의 사람들이 이런 공개적인 발표를 위조할수 있다. 다시말하여 어떤 사용자가 사용자 A이라고 가정하고 공개열쇠를 다른 가입자에게 보내거나 이런 공개열쇠를 광고할수 있다. 사용자 A가 위조를 발견하고 다른 가입자들에게 경고를 울릴 때까지 위조자는 A에게 오는 모든 암호화된 통보문들을 읽을수 있으며 인증을 위한 위조된 열쇠를 리용할수 있다(그림 6-3을 보시오).

공개적으로 리용할수 있는 등록부

공개적으로 리용할수 있는 공개열쇠들의 동적목록을 관리함으로써 보다 큰 보안을 달성할수 있다. 공개열쇠목록의 관리와 배송은 어떤 신용실체나 조직이 책임 져야 한다(그림 6-11). 이런 방식은 다음과 같은 요소들을 포함할것이다.

1. 책임자가 매 가입자에 대하여 {이름, 공개열쇠}실체들로 이루어 진 목록을 관리한다.
2. 매 가입자는 목록책임자에게 공개열쇠를 등록한다. 등록은 공개적이여야 하거나 어떤 형태의 비밀로 인증된 통신에 의해 진행되여야 한다.

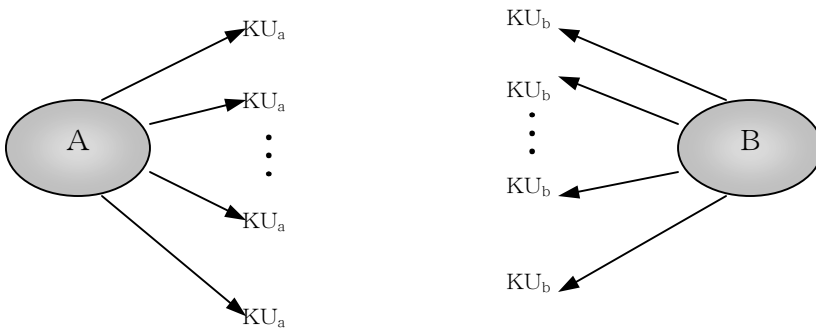



그림 6-10. 조종되지 않은 공개열쇠배포

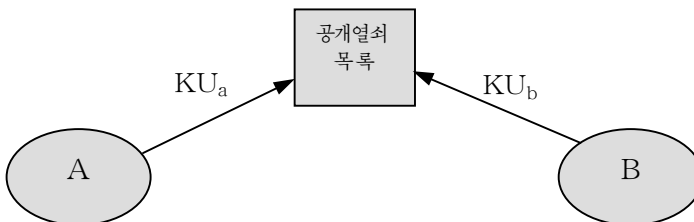



그림 6-11. 공개열쇠공개(발표)

3. 가입자는 임의의 시각에 새로운 열쇠로 이미 존재하는 열쇠를 교체할수도 있다. 왜냐하면 이미 다량의 자료들에 리용된 공개열쇠를 교체할것을 요구하거나 대응하는 비밀열쇠가 어떤 방식에 의해 손상되였기때문이다.
4. 책임자는 주기적으로 목록전부를 공개하거나 목록을 갱신한다. 가령 전화번호책과 같은 하드복사판이 공개되며 광범히 류통되는 신문에 렬거하여 갱신된다.
5. 가입자는 또한 전자적으로 목록을 참조한다. 이 목적을 위하여 책임자로부터 가입자에로의 안전하며 인증된 통신은 명령적이여야 한다.

이 방식은 분명히 개별적인 공개적발표보다 안전하지만 아직 약점들이 있다. 적이 목록책임자의 비밀열쇠를 얻으려고 하거나 계산하려고 한다면 적은 위조된 공개열쇠를 책임적으로 통과시키고 그다음에는 계속하여 어떤 가입자의 역할을 하며 임의의 가입자에게 보낸 통보문을 도청할것이다. 같은 결과를 달성하는 다른 하나의 방법은 적이 책임자가 가지고 있는 레부호를 자의대로 수정하는것이다.

공개열쇠책임자

목록의 공개열쇠를 배송하기 위한 엄격한 규정을 주면 공개열쇠배포를 위한 강한 보안을 달성할수 있다. 전형적인 씨나리오를 [POPE79]의 그림에 기초한 그림 6-12로 보여 준다. 앞에서와 같이 이 씨나리오는 중앙책임자가 모든 가입자들의 공개열쇠들의 동적목록을 관리한다고 가정한다. 그외에도 매 가입자는 책임자의 공개열쇠를 확고히 알며 오직 책임자만이 대응하는 비밀열쇠를 안다. 다음과 같은 단계들이 있다(그림 6-12에서 번호로 표시된것).

1. A는 B의 현재 공개열쇠에 대한 요구를 가지고 있는 공개열쇠책임자에게 일부인 있는 통보문을 보낸다.
2. 책임자는 자기의 비밀열쇠 KR_{auth}을 리용하여 암호화된 통보문으로 응답한다. 이리하여 A는 책임자의 공개열쇠를 리용하여 통보문을 복호화할수 있다. 따라서 A는 통보문이 책임자에 의해 발생되였음을 확신한다. 통보문은 다음과 같은 것들을 포괄한다.

- A가 B에게 보내는 통보문을 암호화할수 있는 B의 공개열쇠 KU_b
- A가 대응하는 앞의 요구와 이 응답을 대조하며 원래의 요구가 책임자에 의해 접수되기전에는 경고가 울리지 않음을 검증하는 원래의 요구
- 원래의 일부인. 그래서 A는 B의 현재 공개열쇠만이 아니라 다른 열쇠도 포함하고 있는 책임자로부터 오는 낡은 통보문이 이 일부인이 아님을 결정할수 있다.

3. A는 B의 공개열쇠를 기억하고 그것을 리용하여 A의 식별자(ID_A)와 이 처리를 유일하게 식별하는데 리용되는 한번쓰기정보(N_1)를 가지고 있는 B에게 통보문을 암호화하여 보낸다.
4,5. B는 A가 B의 공개열쇠를 받는것과 같은 방식으로 책임자로부터 A의 공개열쇠를 받는다.

이 시점에서 공개열쇠는 A와 B에게 안전하게 전달되며 그것들은 자기의 보호된 교환을 시작할것이다. 그러나 다음과 같은 2개의 보충적인 단계들이 요구될수 있다.

6. B는 KU_a로 암호화되고 B에 의해 생성된 새로운 한번쓰기정보(N_2)는 물론 A의 한번쓰기정보(N_1)를 포함하는 통보문을 A에게 보낸다. 오직 B만이 통보문으로 (3)을 복호화하였으므로 통보문으로 (6)에서의 N_1의 출현은 대응자가 B인 A를 담보한다.
7. A는 B의 공개열쇠를 리용하여 B의 대응자가 A인 B를 담보하는 암호화된 N_2를 돌려 준다.

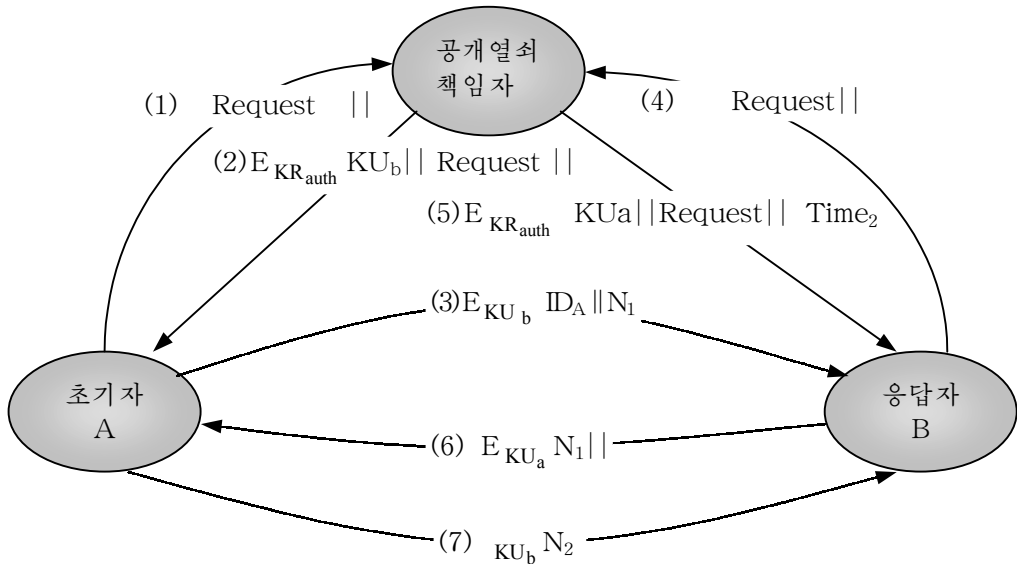



그림 6-12. 공개열쇠배포씨나리오

이리하여 총 7개의 통보문이 요구된다. 그러나 A와 B는 둘 다 다른 사람의 공개열쇠를 보관할수 있으므로 초기 4개의 통보문은 앞으로의 리용을 위해 오직 드물게만 리용되여야 한다. 사용자는 주기적으로 현재성을 담보하는 자기 대응자들의 공개열쇠의 참신한 복사를 요구한다.

공개열쇠확인

그림 6-12의 씨나리오는 매력적이지만 아직 일련의 약점들이 있다. 공개열쇠책임자에게는 체계에서 일련의 애로가 있다. 즉 사용자가 서로 접촉하려는 매 다른 사용자의 공개열쇠를 위해 책임자에게 신청해야 한다. 그러나 앞에서와 마찬가지로 책임자가 관리하는 이름 및 공개열쇠들의 목록은 간섭에 약하다.

콘헬더(Kohnfelder)［KOHN78］가 처음으로 제기한 또 다른 하나의 방식은 열쇠를 공개열쇠책임자로부터 직접 얻는것과 같이 믿을수 있는 방법으로 공개열쇠책임자를 접촉함이 없이 열쇠를 교환할 가입자가 리용할수 있는 **확인**을 리용하는것이다. 매 확인은 공개열쇠와 다른 정보를 포함하고 확인책임자에 의해 창조되며 그것은 대조할 비밀열쇠를 가진 가입자에게 주어 진다. 가입자는 자기의 확인을 전송하는 방법으로 자기열쇠정보를 다른 사람에게 나른다. 다른 가입자는 확인이 책임자에 의해 창조되였음을 검증할수 있다. 이 방식에 관한 다음과 같은 요구들을 고찰할수 있다.

1. 임의의 가입자는 확인소유자의 이름과 공개열쇠를 결정하는 확인을 읽을수 있다.
2. 임의의 가입자는 확인이 확인책임자로부터 발생되였으며 그것이 위조아님을 검증할수 있다.
3. 오직 확인책임자만이 확인을 창조하고 갱신할수 있다.

이런 요구들은 최초에 제기된 [KOHN78]에 의해 만족된다. [DENN83]에서는 다음과 같은 보충적인 요구들을 첨가하였다.

4. 임의의 가입자는 확인의 현재성을 검증할수 있다.

확인방식을 그림 6-13으로 보여 준다. 매 가입자는 공개열쇠를 공급하며 확인을 요구하는 확인책임자에게 신청한다. 신청은 개별적으로 되여야 하거나 어떤 형태의 인증된 통신에 의해 진행되여야 한다. 가입자 A에 대하여 책임자는

$$C_A = E_{KR_{auth}}[T, ID_A, KU_a]$$

형태의 확인을 제공한다. 여기서 KR_{auth}는 책임자가 리용하는 비밀열쇠이다. A는 이때 이 확인을 임의의 다른 가입자에게 통과시킬수 있는데 다른 가입자는 다음과 같은 확인을 읽고 검증할수 있다.

$$D_{KU_{auth}}[C_A] = D_{KU_{auth}}[E_{KR_{auth}}[T, ID_A, KU_a]] = (T, ID_A, KU_a)$$

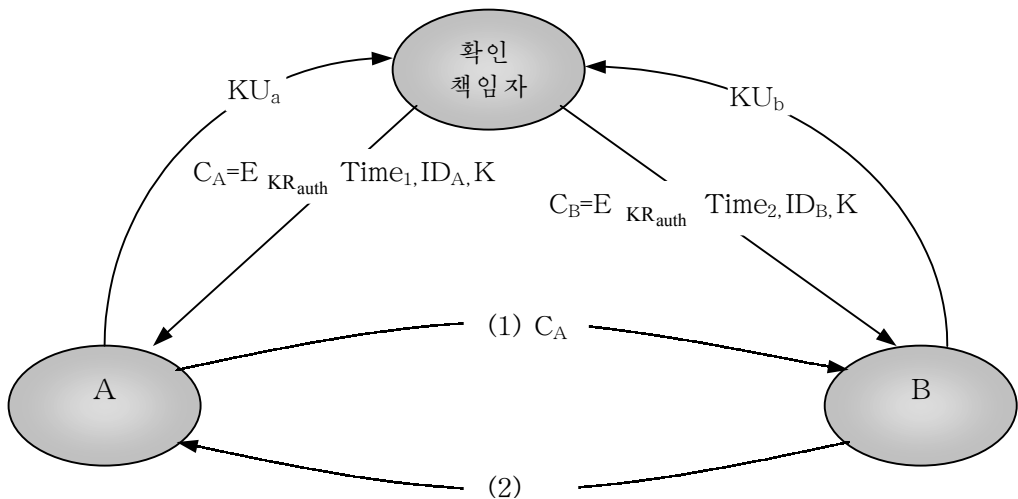



그림 6-13. 공개열쇠확인의 교환

수신자는 책임자의 공개열쇠 KU_{auth}를 리용하여 확인을 복호화한다. 확인은 오직 책임자의 공개열쇠만을 리용하여 읽을수 있으므로 확인이 확인책임자로부터 온다는것을 검증할수 있다. 요소 ID_A와 KU_a는 확인등록부의 이름과 공개열쇠를 수신자에게 공급한다. 최종적으로 일부인된 T는 확인의 현재성을 담보한다. 일부인은 다음과 같은 씨나리오라고 생각한다. A의 비밀열쇠가 적에 의해 알려 진다. A는 새로운 비밀열쇠/공개열쇠쌍을 생성하여 새로운 확인을 위해 확인책임자에게 신청한다. 그동안에 적은 낡은 확인을 B에게 재배치한다. 이때 B가 손상된 낡은 공개열쇠를 리용하여 통보문을 암호화하면 적은 이런 통보문을 읽을수 있다.

이 문맥에서 비밀열쇠의 손상은 신용카드의 잃음과 비교할수 있다. 그 소유자들은 신용카드번호를 취소하지만 모든 통신자들이 낡은 신용카드가 구식이라는것을 알 때까지 위험은 남아 있다. 이리하여 일부인은 만기날자와 같은것으로 종사한다. 확인이 충분히 낡으면 만기가 되였다고 가정한다.

비밀열쇠의 공개열쇠배포

일단 공개열쇠가 배송되였거나 참조가능하게 되였다면 도청(그림 6-2), 간섭(그림 6-3) 또는 둘 다(그림 6-4) 좌절시키는 비밀통신이 가능하다. 그러나 일부 사용자들은 통신에서 공개열쇠암호를 배제하려고 한다. 왜냐하면 상대적으로 자료전송속도가 느리기때문이다. 따라서 공개열쇠암호는 전통암호에 리용되는 비밀열쇠의 배송수단으로서 아주 적합하다.

단순한 비밀열쇠배포

아주 단순한 방식을 그림 6-14에서 보여 준다[MERK79]. A가 B와 통신하려고 한다면 다음과 같은 절차를 채용한다.

1. A는 공개/비밀열쇠쌍{KU_a,KR_a}를 생성하고 KU_a와 A의 식별자 ID_A로 이루어진 통보문을 B에게 전송한다.
2. B는 비밀열쇠 K_S를 생성하고 A의 공개열쇠로 암호화한 그것을 A에게 전송한다.
3. A는 $D_{KR_a}[E_{KU_a}[K_S]]$를 계산하여 비밀열쇠를 구한다. 오직 A만이 통보문을 복호화할수 있으므로 A와 B만이 K_S의 신원을 알것이다.
4. A는 KU_a와 KR_a를 버리며 B는 KU_a를 버린다.

A와 B는 이제는 전통암호와 대화열쇠를 리용하여 안전하게 통신할수 있다. 교환을 완성한 다음에 A와 B는 둘 다 K_S를 버린다. 이것은 단순함에도 불구하고 흥미 있는 규약이다. 통신이 시작되기전에는 열쇠가 존재하지 않으며 통신이 완료된 다음에도 열쇠가 존재하지 않는다. 이리하여 열쇠손상의 위험은 최소로 된다. 동시에 통신은 도청으로부터 안전하다.

이 규약은 능동공격에 약화될수 있다. 적 E가 간섭통신통로를 조종한다면 E는 검출함이 없이 다음과 같은 방식으로 통신을 방해할수 있다.

1. A는 공개/비밀열쇠쌍{KU_a, KR_a}을 생성하고 B에게 보낼 KU_a와 A의 식별자

ID_A로 이루어 진 통보문을 전송한다.

2. E는 통보문을 가로 채여 자기자신의 공개/비밀열쇠쌍{KU_e, KR_e}를 창조하고 $KU_e||ID_A$를 B에게 전송한다.
3. B는 비밀열쇠 K_S를 생성하여 $E_{KU_e}[K_S]$를 전송한다.
4. E는 통보문을 가로 채여 $D_{KR_e}E_{KU_e}[K_S]$를 계산함으로써 K_S를 알아 낸다.
5. E는 $E_{KU_e}[K_S]$를 A에게 전송한다.

결과는 A와 B들이 다 K_S를 알지만 Ks가 E에게 드러나게 된다는것을 모르는것이다. A와 B가 이제는 K_S를 리용하여 통보문을 교환할수 있다. E는 더는 실제적으로 간섭하지 않지만 단순히 도청은 한다. K_S를 알면 E는 통보문으로 모두를 복호화할수 있으며 A와 B는 둘 다 그 문제를 모르고 있다. 이리하여 이 단순한 규약은 오직 도청 당할수 있는 환경속에서만 유용하다.

기밀성과 인증을 가진 비밀열쇠배포

NEED78에서 제시한 방식에 기초한 그림 6-15는 능동공격과 피동공격으로부터 보호된다. A와 B가 앞에서 서술한 방식들중의 하나로 공개열쇠를 교환하였다고 가정했을 때 시작한다. 이때 다음과 같은 단계들이 발생된다.

1. A는 B의 공개열쇠를 리용하여 A의 식별자(ID_A)와 이 처리를 유일하게 식별하는데 리용되는 한번쓰기정보(N_1)를 포함하는 통보문을 암호화하여 B에게 보낸다.
2. B는 B에 의해 생성된 새로운 한번쓰기정보(N_2)는 물론 A의 한번쓰기정보(N_1)를 포함하며 KU_a로 암호화하여 A에게 보낸다. 오직 B만이 통보문 (1)을 복호화하므로 통보문 (2)에서 N_1의 출현은 대응자가 B인 A를 담보한다.

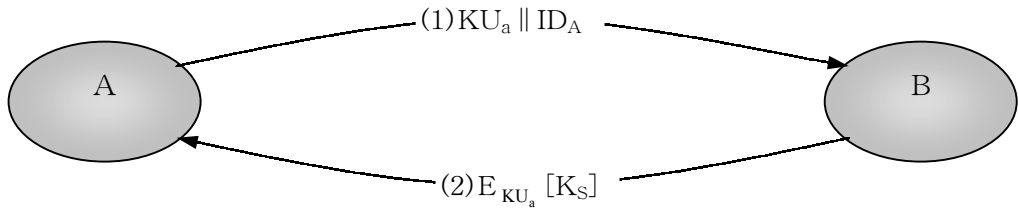



그림 6-14. 대화열쇠를 설정하는 공개열쇠암호의 단순한 리용

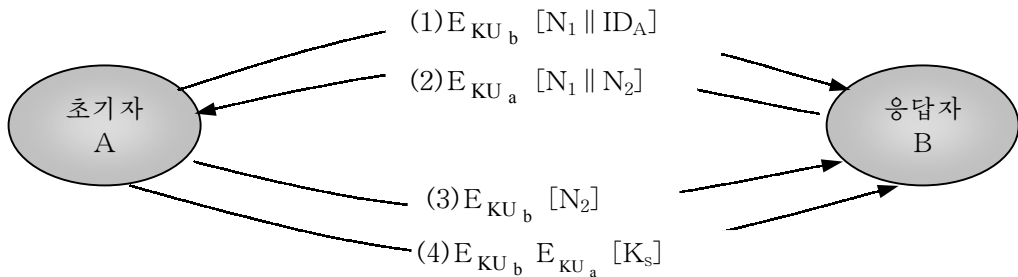



그림 6-15. 비밀열쇠의 공개열쇠배포

3. A는 B의 공개열쇠를 리용하여 대응자가 A임을 B에게 담보하는 암호화된 N_2 을 돌려 준다.
4. A는 비밀열쇠 K_S를 택하고 $M = E_{KU_b}[E_{KR_a}[K_S]]$를 B에게 보낸다. B의 공개열쇠에 의한 이 통보문의 암호화는 오직 B만이 그것을 읽을수 있음을 담보한다. 즉 A의 공개열쇠로 한 암호화는 오직 A만이 그것을 보냈다는것을 담보한다.
5. B는 $D_{KU_a}[D_{KR_b}[M]]$을 계산하여 비밀열쇠를 회복한다.

이 방식의 첫 3개 단계는 그림 6-12의 마지막 3개 단계와 같다는것을 강조한다. 그 결과는 이 방식이 비밀열쇠의 교환에서 기밀성과 인증을 둘 다 담보한다는것이다.

복합방식

비밀열쇠를 배송하는데 공개열쇠암호를 리용하는 다른 또 하나의 방식은 IBM의 주틀 LE93을 리용하는 복합방식이다. 이 방식은 매 사용자와 비밀주열쇠를 공유하여 주열쇠로 암호화된 비밀대화열쇠를 배송하는 열쇠배포쎈터(KDC)를 리용한다. 공개열쇠방식은 주열쇠를 배송하는데 리용된다. 다음과 같이 리상적인것은 이 3준위방식을 리용하는데 제공된다.

- **성능:** 대화열쇠를 자유로이 변경하는 많은 응용에는 특히 처리지향응용들이 있다. 공개열쇠암호에 의한 대화열쇠의 배송은 전면적인 체계의 성능을 저하시킨다. 왜냐하면 공개열쇠암호화와 복호화는 상대적으로 높은 부하를 가지기때문이다.
  3준위계층에서 공개열쇠암호는 오직 사용자와 KDC사이의 주열쇠를 갱신하는데만 때때로 리용된다.
- **거꾸로의 호환성:** 복합방식은 최소의 분렬 또는 쏘프트웨어변경을 가진 존재 KDC방식우에서 쉽게 감추어 진다.

공개열쇠층의 보충은 주열쇠를 배송하는 안전하고 효과적인 수단을 제공한다. 이것은 하나의 KDC가 널리 분산된 사용자들의 모임을 봉사하는 배치구성에서 우월하다.

6.4 디피-헬만열쇠교환

처음으로 발표된 공개열쇠알고리듬은 공개열쇠암호를 정의한 디피-헬만의 론문 [DIFF76b]에서 제기되였으며 일반적으로 디피-헬만열쇠교환이라고 한다. 일련의 상업적제품들은 이 열쇠교환기술을 채용한다.

알고리듬은 그 효과성을 위하여 리산로그계산의 복잡성에 의거한다. 간단히 보면 리산로그를 다음과 같은 방법으로 정의할수 있다. 우선 씨수 p의 원시뿌리를 정의하는데 이것은 그의 제곱이 1부터 p-1사이의 모든 옹근수들을 생성한다. 즉 a가 씨수 p의 원시뿌리이면 수

$$a \bmod p,\ a^2 \bmod p,\ \cdots\cdots,\ a^{p-1} \bmod p$$

는 모두 서로 다르며 어떤 치환에 의하여 1부터 p-1사이의 옹근수들로 이루어 진다.

임의의 옹근수 b와 씨수 p의 원시뿌리 a에 대하여

$$b = a^i \bmod p \qquad \text{여기서 } 0 \le i \le (p-1)$$

인 유일한 제곱지수 i를 구할수 있다. 제곱지수 i를 mod p에 관하여 밑수 a인 b의 리산로그 또는 첨수라고 하고 $\mathrm{ind}_{a,p}(b)$로 표시한다. 보다 구체적인것은 7장을 보시오.

이런 환경하에서 그림 6-16에서 개괄한 디피-헬만열쇠교환을 정의할수 있다. 이 방식에 대하여 2개의 공개적으로 알려 진 수 즉 씨수 q와 q의 원시뿌리인 α가 있다. 사용자 A와 B가 열쇠를 교환하려고 한다고 가정하자. 사용자 A는 우연옹근수 $X_A<q$를 택하여 $Y_A = \alpha^{X_A} \bmod q$를 계산한다. 마찬가지로 사용자 B는 독립적으로 우연옹근수 $X_B<q$를 택하여 $Y_B = \alpha^{X_B} \bmod q$를 계산한다. 매측은 X값을 비밀로 하며 Y값을 다른 측에 공개하도록

대역적공개요소

q	씨수
α	$\alpha<q$이고 α는 q의 원시뿌리

사용자 A의 열쇠생성

비밀열쇠 X_A를 택한다.	$X_A<q$
공개열쇠 Y_A를 계산한다.	$Y_A = \alpha^{X_A} \bmod q$

사용자 B의 열쇠생성

비밀열쇠 X_B를 택한다.	$X_B<q$
공개열쇠 Y_B를 계산한다.	$Y_B = \alpha^{X_B} \bmod q$

사용자 A에 의한 비밀열쇠생성

$K = (Y_B)^{X_A} \bmod q$

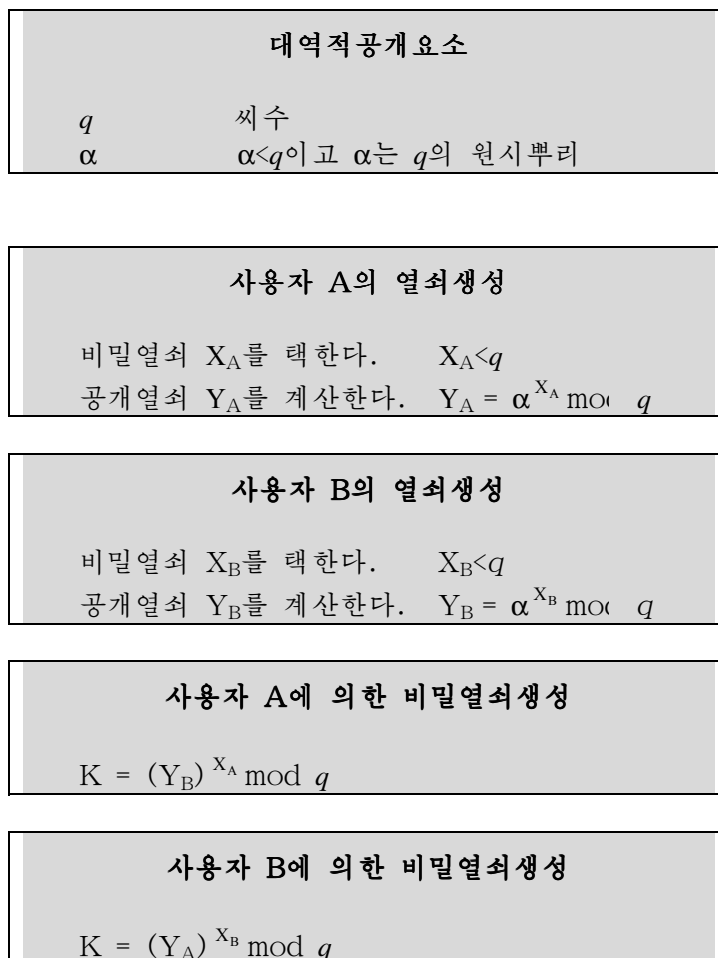
사용자 B에 의한 비밀열쇠생성

$K = (Y_A)^{X_B} \bmod q$

그림 6-16. 디피-헬만열쇠교환알고리듬

한다. 사용자 A는 열쇠를 K = $(Y_B)^{X_A}$ mod q로, 사용자 B는 열쇠를 K = $(Y_A)^{X_B}$ mod q로 계산한다. 이 두 계산은 같은 결과를 준다. 즉

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q \qquad \text{모드산수규칙으로부터} \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

이리하여 두 측은 비밀열쇠를 교환하였다. 더 나아가서 X_A와 X_B는 비밀이므로 적은 오직 다음과 같은 요소 q, a, Y_A와 Y_B만을 가진다. 따라서 적은 열쇠를 구하기 위하여 리산로그를 취하는데 집중하게 된다. 실례로 사용자 B의 비밀열쇠를 공격하려면 적은

$$X_B = \mathrm{ind}_{a,p}(Y_B)$$

를 계산해야 한다. 그래야 적은 사용자 B가 계산하는것과 같은 방식으로 열쇠 K를 계산할수 있다.

디피-헬만열쇠교환의 보안은 모드씨수에 관한 제곱지수를 계산하는것이 상대적으로 쉽다면 리산로그를 계산하는것은 아주 힘들다는 사실에 기초한다. 큰 씨수에 대하여 후자의 과제는 불가능한것으로 고찰된다.

여기에는 하나의 실례가 있다[SEBE89]. 열쇠교환은 씨수 $q=97$과 97의 원시뿌리 $\alpha=5$의 리용에 기초한다. A와 B는 비밀열쇠 $X_A=36$과 $X_B=58$을 택한다. 매개는 다음과 같은 공개열쇠를 계산한다.

$$
\begin{aligned}
Y_A &= 5^{36} = 50 \bmod 97 \\
Y_B &= 5^{58} = 44 \bmod 97
\end{aligned}
$$

공개열쇠들을 교환한 다음에 매개는 다음과 같은 공통비밀열쇠를 계산할수 있다.

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod 97 = 44^{36} = 75 \bmod 97 \\
K &= (Y_A)^{X_B} \bmod 97 = 50^{58} = 75 \bmod 97
\end{aligned}
$$

{50, 44}로부터 공격자는 75를 쉽게 계산할수 없다.

그림 6-17은 디피-헬만계산방법을 리용하는 단순한 규약을 보여 준다. 사용자 A가 사용자 B와 련결하고 비밀열쇠를 리용하여 그 련결우에서 통보문을 암호화하려고 한다고 가정하자. 사용자 A는 비밀열쇠 X_A를 생성하고 Y_A를 계산하여 그것을 사용자 B에게 보낼수 있다. 사용자 B는 비밀값 X_B를 생성하고 Y_B를 계산하여 사용자 A에게 보내는것으로 응답한다. 두 사용자가 이제는 열쇠를 계산할수 있다. 필요한 공개값 q와 α는 미리 알려 져 있어야 한다. 다른 한가지로서 사용자 A는 q와 α에 대한 값을 취하고 첫 통보문속에 이것을 포함시킨다.

디피-헬만알고리듬리용의 다른 실례로서 사용자들의 집단(즉 LAN의 모든 사용자)의 매개가 긴 비밀값 X_A를 생성하고 공개값 Y_A를 계산한다고 가정하자. 이 공개값들은 q와 α에 대한 대역적공개값들과 함께 어떤 중심목록에 기억된다. 임의의 시각에 사용자 B는 사용자 A의 공개값을 참조하여 비밀열쇠를 계산할수 있으며 그것을 리용하여 암호화된 통보문을 사용자 A에게 보낼수 있다. 중심목록이 믿을만하면 이런 형태의 통신은 기밀성과 인증을 제공한다. A와 B만이 열쇠를 결정할수 있으므로 다른 사용자는 통보문을 읽을수 없다(기밀성). 수신자 A는 이 열쇠를 리용하여 오직 사용자 B만이 통보문을 창조함을 안다(인증). 그러나 이 기술은 재차공격에는 보호되지 못한다.

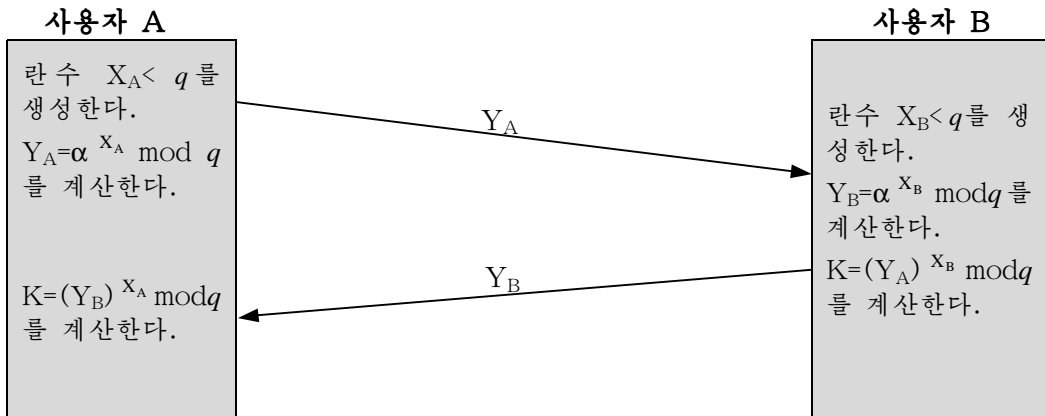



그림 6-17. 디피-헬만열쇠교환

6.5 타원곡선암호

암호화/복호화와 수자서명을 위하여 공개열쇠를 리용하는 제품과 표준들의 대부분은 RSA를 리용한다. 본바와 같이 안전한 RSA의 비트길이는 최근년간 늘어 나며 이것은 RSA를 리용하는 무거운 부담으로 되고 있다. 이 부담은 특히 큰 수의 비밀처리를 요구하는 전자상거래같은데서 혼란을 일으키고 있다. 최근 경쟁체계 즉 타원곡선암호(ECC)같은것들이 RSA에 도전하기 시작하였다. 이미 ECC는 표준화효과를 보여 주고 있으며 공개열쇠암호를 위한 IEEE P1363표준을 포함하고 있다.

RSA에 비하여 ECC는 원리적측면에서 훨씬 적은 비트크기로 동등한 보안을 보장하므로 보다 간결하다. 다른 한편 ECC의 리론은 최근시기에 제기되여 제품이 나오기 시작하였으며 그 약점이 암호분석자들에 의해 엄밀하게 조사되고 있다. 이리하여 ECC의 신용수준은 아직 RSA만큼은 높지 못하다.

ECC는 기본적으로 RSA나 디피-헬만을 설명하는것보다 더 힘들며 완전한 수학적서술은 이 책의 범위를 벗어 난다. 이 절에서는 타원곡선에 대한 일련의 개념과 ECC를 제기한다.

타원곡선

타원곡선은 타원이 아니다. 그것을 그렇게 부르게 된 리유는 타원주를 계산하는데 리용된 방정식과 류사한 3차방정식으로 서술되기때문이다. 일반적으로 타원곡선

의 3차방정식은

$$y^2+axy+by=x^3+cx^2+dx+e$$

의 형태를 취한다. 여기서 a, b, c, d 및 e는 어떤 단순한 조건을 만족시키는 실수이다. 또한 임의의 타원곡선의 정의에 포함된것은 0으로 표시한 하나의 원소이며 그것을 무한원점 또는 령점이라고 부르는데 앞으로 론의하게 된다. 이런 방정식을 3차방정식이라고 한다. 왜냐하면 그 방정식의 최고제곱지수가 3이기때문이다. 그림 6-18은 타원곡선의 두 실례를 보여 준다. 보는바와 같이 이 식은 때때로 이상한 곡선을 나타낸다.

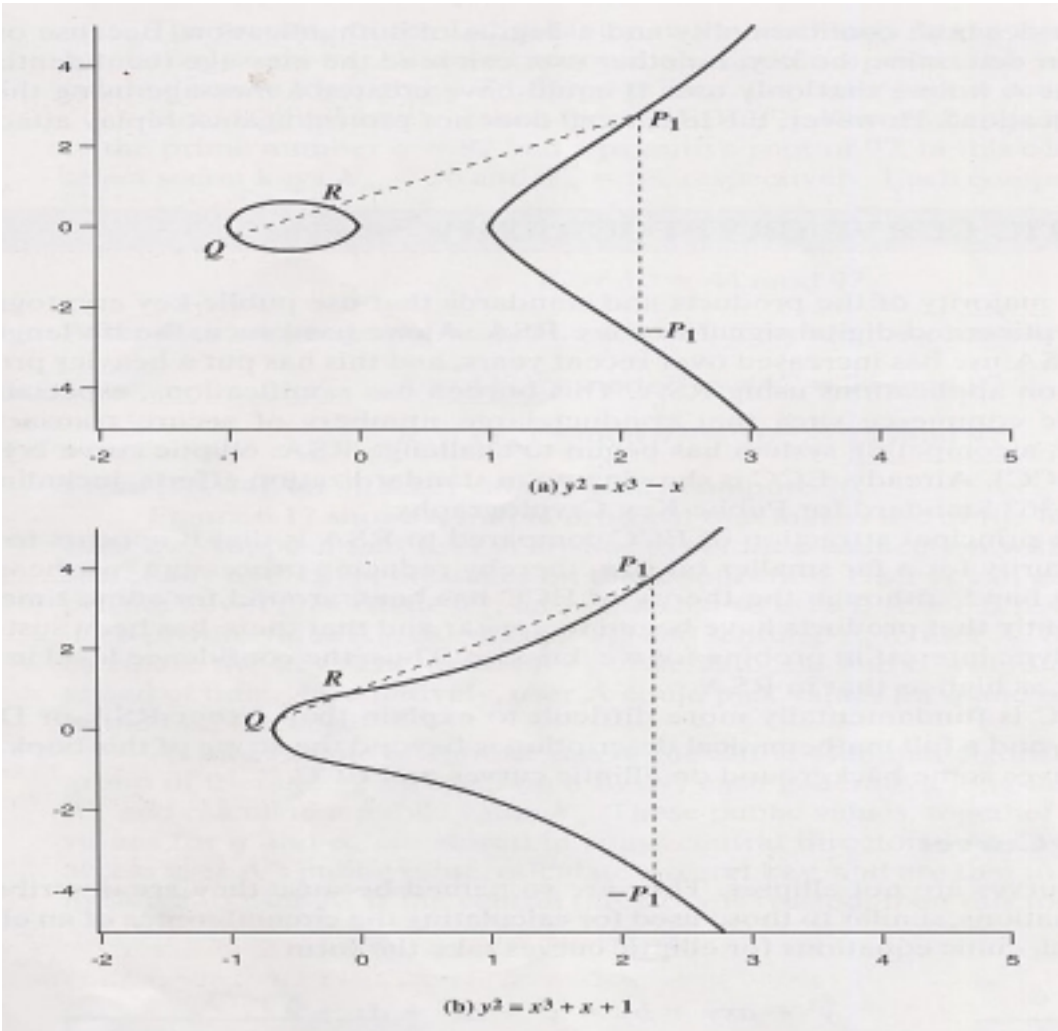



그림 6-18. 타원곡선의 실례

타원곡선우에서 더하기연산을 다음과 같은 간단한 형태로 정의할수 있다. 타원곡선우의 세 점이 한 직선우에 놓이면 그 합은 0이다. 이 정의로부터 타원곡선우에서 더하기규칙을 다음과 같이 정의할수 있다.

1. O는 더하기에 관한 령원소로서 종사한다. 이리하여 O=$-O$이며 타원곡선우의 임의의 점 P에 대하여 $P+O=P$이다.
2. 수직선은 같은 x자리표를 가진 두 점 즉 $P_1=(x, y)$와 $P_2=(x, -y)$에서 곡선을 만난다. 또한 무한원점에서 곡선을 만난다. 그러므로 $P_1+P_2+O=O$이고 $P_1=-P_2$이다. 따라서 점의 반대점은 같은 x자리표를 가지지만 반대의 y자리표를 가지는 점이다. 그림 6-18에 이것을 보여 주었다.
3. 서로 다른 x자리표를 가지는 두 점 Q와 R를 더하기 위하여 그들사이의 직선을 그어 제3의 점인 사귐점 P_1를 구한다. 사귐점인 유일한 점 P_1가 있다는것을 쉽게 볼수 있다(직선이 $P_1=Q$이거나 $P_1=R$를 취한 경우에 각각 Q 또는 R에서 곡선에 대한 탕겐스가 아니라면). 이때 $Q+R+P_1=O$이며 따라서 $Q+R=-P_1$임을 강조한다. 그림 6-18은 이 구조를 보여 준다.
4. 점 Q를 두번 더하기 위하여 탕겐스선을 긋고 다른 사귐점을 구한다. 이때

$$Q+Q=2Q=-S$$

앞에서 말한 더하기규칙은 가환성과 결합성과 같은 더하기의 표준속성들을 만족시킨다. 정의옹근수 k에 의한 타원곡선우의 점 P의 곱하기는 P의 k번 합으로 정의한다. 이리하여 $2P=P+P$, $3P=P+P+P$ 등이다.

유한체우에서의 타원곡선

ECC에서는 유한체우에서 정의된 제한된 형태의 타원곡선을 취급한다. 암호학에 특별히 관심을 가지게 되는것은 mod p에 관한 타원군이라는것이다. 여기서 p는 씨수이다. 이것은 다음과 같이 정의된다.

$$4a^3+27b^2 \pmod{p} \neq 0$$

인 p보다 작은 두 부아닌 옹근수 a와 b를 택한다. 이때 $E_p(a, b)$는 무한원점 O와 함께

$$y^2 \equiv x^3+ax+b \pmod{p} \qquad (6\text{-}1)$$

를 만족시키는 p보다 작은 부아닌 옹근수들의 쌍인 mod p에 관한 타원군을 표시한다.

실례로 $p=23$이라고 하고 타원곡선 $y^2=x^3+x+1$을 고찰하자. 이 경우에 $a=b=1$이다. mod 23에 관한 조건을 만족시키는 $4\times1^3+27\times1^2 \pmod{23}=8\neq0$이 있다.

우의 항목에서 방정식은 그림 6-18의 ㄴ과 같다. 이 그림은 그 방정식을 만족시키는 실수점모두를 가지는 련속곡선을 보여 준다. 타원군에서는 오직 mod p에 관한 방정식을 만족시키는 $(0, 0)$부터 (p, p)까지의 원주우의 1-4분구의 부아닌 옹근수들에만 관심을 가진다. 표 6-4는 $E_{23}(1, 1)$의 부분인 (0아닌)점들을 렬거한다. 일반적으로 다음과 같은 방법으로 렬거한다.

1. $0 \le x < p$인 매 x에 대하여 $x^3 + ax + b \pmod{p}$를 계산한다.
2. 앞단계의 매 결과에 대하여 그것이 mod p에 관하여 두제곱뿌리를 가지는가를 결정한다. 아니라면 이런 값 x를 가지는 점이 $E_p(a,b)$에는 없다. 그렇다면 두제곱뿌리연산을 만족시키는 y의 2개 값이 있을것이다(그 값이 0인 하나의 y값이 아니라면). 이런 (x,y)값들은 $E_p(a,b)$안의 점들이다.

$E_p(a,b)$우에서의 더하기에 관한 규칙에는 그림 6-18에서 보여 준 기하학적기술이 대응한다. 그것을 다음과 같이 모든 점 $P, Q \in E_p(a,b)$에 대하여 정식화할수 있다.

1. $P + O = P$
2. $P = (x, y)$이면 $P + (x, -y) = O$. 점 $(x, -y)$는 $-P$로 표시되는 P의 반대점이다. $(x, -y)$는 그라프에서(그림 6-18의 ㄴ) 그리고 $E_p(a, b)$에서 보여 주는바와 같이 타원곡선우의 점이다. 실례로 $E_{23}(1,1)$에서 $P = (13,7)$에 대하여 $-P = (13,-7)$이다. 그러나 -7 mod 23=16. 따라서 $-P = (13,16)$이고 역시 $E_{23}(1,1)$에 속한다.

표 6-4. 타원곡선 $E_{23}(1,1)$우의 점

(0,1)	(6,4)	(12,19)
(0,22)	(6,19)	(13,7)
(1,7)	(7,11)	(13,16)
(1,16)	(7,12)	(17,3)
(3,10)	(9,7)	(17,20)
(3,13)	(9,16)	(18,3)
(4,0)	(11,3)	(18,20)
(5,4)	(11,20)	(19,5)
(5,19)	(12,4)	(19,18)

3. $P = (x_1, y_1)$, $Q = (x_2, y_2)$이며 $P \neq -Q$이면 $P + Q = (x_3, y_3)$은 다음과 같은 규칙에 의해 결정된다.

$$x_3 \equiv \lambda^2 - x_1 - x_2 \pmod{p}$$
$$y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod{p}$$

여기서

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & P \neq Q\text{일 때} \\[2ex] \dfrac{3x_1^2 + a}{2y_1} & P = Q\text{일 때} \end{cases}$$

두개의 실례를 보자[JURI97]. $P = (3,10)$이고 $Q = (9,7)$이라고 하자. 이때

$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = \frac{-1}{2} \equiv 11 \mod 23$$
$$x_3 = 11^2 - 3 - 9 = 109 \equiv 17 \mod 23$$
$$y_3 = 11(3-(-6)) - 10 = 89 \equiv 20 \mod 23$$

이리하여 $P+Q=(17,20)$. $2P$를 구하기 위하여

$$\lambda = \frac{3(3^2)+1}{2\times10} = \frac{5}{20} = \frac{1}{4} \equiv 6 \mod 23$$
$$x_3 = 6^2 - 3 - 3 = 30 \equiv 7 \mod 23$$
$$y_3 = 6(3-7) - 10 = -34 \equiv 12 \mod 23$$

이며 $2P=(7,12)$이다. 다시 곱하기는 더하기의 반복으로 정의된다. 실례로 $4P=P+P+P+P$ 이다.

타원곡선암호

ECC의 더하기연산은 RSA의 모드곱하기와 같으며 다중더하기는 모드제곱연산과 같다. 타원곡선을 리용하여 암호체계를 구성하기 위해서는 두 씨수의 적을 인수분해하거나 리산로그를 취하는데 대응하는 《힘든 문제》를 구해야 한다.

방정식 $Q=kP$를 고찰하자. 여기서 Q, $P\in E_p(a,b)$이고 $k < p$이다. k와 P가 주어 졌을 때 Q를 계산하는것은 상대적으로 쉽지만 Q와 P가 주어 졌을 때 k를 구하는것은 상대적으로 힘들다.

이 항목에서는 이 기술의 요지를 주는 ECC에 대한 두 방식을 보여 준다.

디피-헬만열쇠교환과의 류사성

타원곡선을 리용한 열쇠교환을 다음과 같은 방법으로 할수 있다. 우선 식 6-1에 대한 타원곡선파라메터 a와 b 그리고 씨수 $p \approx 2^{180}$을 취한다. 이것은 점들의 타원군 $E_p(a,b)$를 정의한다. 다음으로 $E_p(a,b)$의 생성기점 $G=(x_1,y_1)$를 취한다. G를 선택하는 중요한 판정기준은 $nG=O$인 n의 최소값이 아주 큰 씨수이라는것이다. $E_p(a,b)$와 G는 모든 가입자들에게 알려 진 암호체계의 파라메터들이다.

사용자 A와 B사이에 열쇠교환은 다음과 같이 진행된다.

1. n보다 작은 옹근수 n_A를 A가 택한다. 이것은 A의 비밀열쇠이다. 이때 A는 공개열쇠 $P_A=n_A\times G$를 생성한다. 공개열쇠는 $E_p(a,b)$의 점이다.
2. 마찬가지로 B도 비밀열쇠 n_B를 택하고 공개열쇠 P_B를 계산한다.
3. A는 비밀열쇠 $K=n_A\times P_B$를 생성한다. B는 비밀열쇠 $K=n_B\times P_A$를 생성한다.

단계 3의 두 계산은 같은 결과를 준다. 왜냐하면

$$n_A\times P_B=n_A\times(n_B\times G)=n_B\times(n_A\times G)=n_B\times P_A.$$

이 방식을 파괴하기 위하여 공격자는 힘든것으로 가정되는 G와 kG가 주어 졌을 때 k를 계산할수 있어야 한다.

실례로 $p=211$과 $E_p(0,-4)$를 취하자. 이때 곡선은 $y^2=x^3-4$이고 $G=(2,2)$이다.

$241G=O$임을 계산할수 있다. A의 비밀열쇠는 $n_A=121$이고 따라서 A의 공개열쇠는 $P_A=121(2,2)=(115,48)$이다. B의 비밀열쇠는 $n_B=203$이므로 B의 공개열쇠는 $203(2,2)=(130,203)$이다. 공유된 비밀열쇠는 $121(130,203)=203(115,48)=(161,169)$이다.

비밀열쇠는 수들의 쌍임을 강조한다. 이 열쇠가 전통암호의 대화열쇠로 리용된다면 하나의 수가 생성되여야 한다. 단순히 x자리표 또는 x자리표에 관한 어떤 단순한 함수를 리용한다.

타원곡선암호화/복호화

타원곡선을 리용하여 암호화/복호화하는 일련의 방식들이 문헌들에서 해석되였다. 여기서는 가장 단순한것을 보게 된다. 이 체계에서 첫 과제는 $x-y$점 P_m으로 보내게 되는 평문통보문 m을 암호화하는것이다. 이것이 암호문으로 암호화되고 계속하여 복호화되는 점 P_m이다. 통보문을 단순히 점의 x자리표나 y자리표로 암호화할수 없다는것을 강조한다. 왜냐하면 일부 자리표들은 $E_p(a,b)$에 없기때문이다. 실례로 표 6-4를 보자. 이 암호화에 대한 몇가지 방식이 있는데 여기서는 언급하지 않지만 그것을 만족시키는 상대적으로 간단한 기술이 있다는것을 다시 강조한다.

열쇠교환체계에서처럼 암호화/복호화체계는 파라메터로서 점 G와 타원군 $E_p(a,b)$를 요구한다. 매 사용자 A는 비밀열쇠 n_A를 택하고 공개열쇠 $P_A=n_A\times G$를 생성한다.

통보문 P_m을 암호화하여 B에게 보내기 위하여 A는 우연옹근수 k를 택하고 점쌍

$$C_m=\{kG,\ P_m+kP_B\}$$

들로 이루어 진 암호문 C_m을 만든다. A는 B의 공개열쇠 P_B를 리용하였다는것을 강조한다. 암호문을 복호화하기 위하여 B는 쌍의 첫번째 점을 B의 비밀열쇠로 곱하고 그 결과를 두번째 점으로 던다. 즉

$$P_m+kP_B-n_B(kG)=P_m+k(n_BG)-n_B(kG)=P_m$$

A는 평문 P_m에 kP_B를 더함으로써 P_m을 마스크한다. A를 제외한 그 누구도 k의 값을 모르므로 P_B가 공개열쇠라고 할지라도 누구도 마스크 kP_B를 제거할수 없다. 그러나 A역시 비밀열쇠 n_B를 안다면 마스크를 충분히 제거할수 있다는 《단서》를 가진다. 통보문을 얻기 위하여 공격자는 힘들게 가정된 G와 kG가 주어 졌을 때 k를 계산해야 한다.

암호화처리의 실례[KOBL94]로서 $p=751$, $E_p(-1,188)$를 취하자. 여기서 곡선 $y^2=x^3-x+188$이고 $G=(0,376)$이다. 타원점 $P_m=(562,201)$로 암호화되고 A가 란수 $k=386$을 택한 통보문을 B에게 보내려고 한다고 가정하자. B의 공개열쇠는 $P_B=(201,5)$이다. $386(0,376)=(676,558)$이고 $(562,201)+386(201,5)=(385,328)$이다. 이리하여 A는 암호문$\{(676,558),(385,328)\}$을 보낸다.

타원곡선암호의 보안

ECC의 보안은 kP와 P가 주어 졌을 때 k를 구하는것이 얼마나 어려운가에 관계된다. 이것을 타원곡선로그문제라고 한다. 타원곡선로그를 취하기 위한 알려 진 가장 빠른 기술은 폴라드 로(Pollard rho)방법이다. 표 6-5는 이 방법과 일반수채채법을 리용하여 주어진 수를 두개의 씨수로 인수분해하는 방법의 효과성을 비교한다. 보는바와 같이 RSA에 비하여 ECC는 상당히 작은 열쇠크기를 리용할수 있다. 더 나아가서 같은 열쇠길이에 대

하여 ECC와 RSA에 요구된 계산량적효과는 비교할수 있다[JURI97]. 이리하여 비교적 안전한 RSA보다 더 짧은 열쇠길이를 가지는 ECC의 리용은 계산량적인 발전이다.

표 6-5. RSA에 비한 타원곡선암호의 분석을 위한 계산량적효과

열쇠크기	MIPS년
150	3.8×10^{10}
205	7.1×10^{18}
234	1.6×10^{28}

ㄱ) 폴라드 로방법을 리용한 타원곡선알고리듬

열쇠크기	MIPS년
512	3×10^{4}
768	2×10^{8}
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

ㄴ) 일반화된 수채채법을 리용한 옹근수 인수분해

참고문헌

3장에서 제시한 참고문헌들은 전통암호는 물론 공개열쇠암호도 포함한다.

[DIFF88]은 2개의 비밀열쇠알고리듬을 계획하는 일련의 시도들을 구체적으로 서술한다. 공개열쇠암호에 대한 함축된 론의는 [NECH93]에서 준다. 구체적인 론의는 [SALO96]에 있다. [CORM90]은 RSA에 대하여 간결하지만 완전하고 리해하기 쉬운 알고리듬들을 준다.

타원곡선암호에 대한 좋은 책은 문헌[MENE93]이다.또한 보다 간결하게 서술한 문헌들은 [KUMA98],[STIN95],[KOBL94]이다.

CORM90 Cormen,T.:Leiserson,C.;and Rivest,R. *Introduction* to *Algorithms*. Cambridge, MA:MITPress,1990.

DIFF88 Diffie, W. •The First Ten Years of Public-Key Cryptography. •*Proceedings of the IEEE*, May 1988. Reprinted in SIMM92.

KOBL94 Koblitz, N. A Course *in Number Theory and Cryptography*. New York: Springer-Verlag,1994.

KUMA98 Kumanduri, R., and Romero, C. *Number Theory with Computer Applications*.Upper Saddle River, NJ:Prentice Hall,1998.

MENE93 Menezes,A. *Elliptic Curve Public Key Cryptosystems*. Boston:Kluwer Academic Publishers,1993.

NECH92 Nechvatal,J.•Public Key Cryptography.홀수0n SIMM92.

SALO96 Salomaa, A. *Public-Key Cryptography*. New York:Springer-Verlag, 1996.

SIMM92 Simmons, G., ed. *Contemporary Cryptology*: *The Science of Information Integrity*. Piscataway, NJ:IEEE Press,1992.

STIN95 Stinson, D. *Cryptography*: *Theory and Practice*.Boca Raton, FL:CRC Press,1995.

문 제

1. RSA와 같은 임의의 특정한 공개열쇠방식의 발견에 앞서서 공개열쇠암호가 리론적으로 가능함을 보여 주는것이 목적인 존재성증명이 밝혀 졌다. 함수 $f_1(x_1)=z_1$; $f_2\ (x_2,y_2)=z_2$; $f_3(x_3,y_3)=z_3$을 고찰하자. 여기서 모든 값들은 $1\le x_i,y_i,z_i\le N$인 옹근수이다. 함수 f_1는 k번째 입력이 $f_1(k)$의 값이고 길이 N인 벡토르 M1에 의해 표현될수 있다. 마찬가지로 f_2과 f_3을 $N\times N$형행렬 M2와 M3으로 표현할수 있다. 목적은 아주 큰 N의 값을 가진 표검색에 의해 암호화/복호화처리를 표현하는것이다. 이런 표는 실행할수 없을 정도로 거대하지만 원리적으로는 구성된다. 이 방식은 다음과 같이 동작한다. 1과 N사이의 옹근수모두의 우연치환으로 M1을 구성한다. 즉 매 옹근수는 M1에 정확히 한번 나타난다. M2도 그렇게 구성하여 매행은 첫 N개 옹근수의 우연치환을 포함한다. 최종적으로는 다음의 조건을 만족시키는 M3에 채운다. $1\le k,p\le N$인 모든 k,p에 대하여

   $f_3(f_2(f_1(k),p),k)=p$　　　　모든 k, p에 대하여 $1\le k,p\le N$일 때

   말로 표현하면
   1) M1은 입구 k를 취하여 출구 x를 내보낸다.
   2) M2는 x와 p를 넣고 z를 내보낸다.
   3) M3은 z와 k를 넣고 p를 내보낸다.

   일단 구성된 3개의 표를 공개한다.

   ㄱ) 선행한 조건을 만족시키는 M3을 구성하는것이 가능하다는것은 분명하다. 실례로 다음과 같이 단순한 경우에 M3안에 요소들을 채우시오.

M1=

5
4
2
3
1

M2=

5	2	3	4	1
4	2	5	1	3
1	3	2	4	5
3	1	4	2	5
2	5	3	4	1

M3=

   약속:M1의 i번째 원소에는 $k=i$를 대응시킨다. M2의 i번째 행에는 $x=i$를 대응시킨다. M2의 j번째 렬에는 $p=j$를 대응시킨다. M3의 i번째 행에는 $z=i$를 대응시키고 M3의 j번째 렬에는 $k=j$를 대응시킨다.

   ㄴ) 두 사용자사이에 암호화와 복호화를 진행하는데 이 표들의 모임을 리용하여 서술하시오.

   ㄷ) 이것은 비밀방식임을 론증하시오.

2. 그림 6-6에서처럼 RSA알고리듬을 리용하여 다음과 같이 암호화와 복호화를 진행하라.

ㄱ) $p=3; q=11; d=7; \mathbf{M}=5$
ㄴ) $p=5; q=11; e=3; \mathbf{M}=9$
ㄷ) $p=7; q=11; e=17; \mathbf{M}=8$
ㄹ) $p=11; q=13; e=11; \mathbf{M}=7$
ㅁ) $p=17; q=31; e=7; \mathbf{M}=2$

암시: 복호화는 생각하는것보다 힘들지 않다. 어떤 좋은 착상을 리용하시오.

3. RSA를 리용하는 공개열쇠체계에서 공개열쇠가 $e=5, n=35$인 사용자에게 보낸 암호문 $C=10$을 분석한다. 평문 M은 무엇인가?
4. RSA체계에서 주어 진 사용자의 공개열쇠가 e=31, n=3599이다. 이 사용자의 비밀열쇠는 무엇인가?
5. RSA알고리듬을 리용하는데서 어떤 작은 회수의 반복적인 암호화가 다시 평문으로 돌아 온다면 좋은것은 무엇인가?
6. RSA로 암호화된 블로크가 있는데 비밀열쇠를 모른다고 가정하자. $n=pq$, e는 공개열쇠라고 하자. 이것은 우리에게 어떤 방식으로 도움을 주는가?
7. RSA공개열쇠암호방식에서 매 사용자에게는 공개열쇠 e와 비밀열쇠 d가 있다. 어떤 사용자가 자기의 비밀열쇠를 루설했다고 가정하자. 새로운 모드를 생성하는것보다 오히려 사용자는 새로운 공개열쇠와 새로운 비밀열쇠를 생성하기로 결심하였다. 이것이 안전한가?
8. 《여보게, 홈스(Holmes),》라고 부른 와트손(Watson)은 열정적으로 다음과 같이 말하였다. 《당신이 최근에 진행한 망안전기술이 나의 암호에 대한 흥미를 돋구어 주었네. 바로 어제 나는 한시각용(one-time pad) 암호를 실행시킬 방도를 찾아 내였네.》

   《아, 정말?》 홈스의 얼굴에서 졸음기가 사라 졌다. 《그래서 자네가 암호학적으로 강도가 높은 렬을 결정방식으로 생성시킬 방도를 찾아 냈단말이지?》

   《그렇네, 홈스. 그 착상은 아주 간단하네. 주어 진 한방향함수 F에 대하여 나는 F를 일정한 표준인수렬에 적용하여 원소들의 긴 모조란수렬을 생성하지. 암호분석자가 F와 렬의 일반적속성을 안다고 가정하는데 그것은 S, S+1, S+2, …처럼 간단하겠지만 S는 비밀이 아니네. 그리고 F의 한방향성으로 인하여 어떤 i에 대하여 $F(S+i)$가 주어 지면 누구도 S를 추출할수 없네, 이렇게 되여 그가 일정한 렬의 토막을 얻어 낸다고 해도 그는 나머지를 결정할수 없게 되네.》

   《와트손, 자네의 생각에는 결함이 없어 보이지만 적어도 F에 의하여 만족되는 어떤 보조적인 조건들이 있어야겠네. 실례로 RSA암호화함수 즉 $F(M)=M^R \bmod N$, K가 비밀이라고 생각해 보라구. 이 함수는 한방향이라고 믿어 지지만 나는 그 리용을 례를 들어 렬 $M=2,3,4,5,6,\cdots$에다 적용하고 싶지 않구만.》

   《홈스, 어째서?》 와트손은 미처 리해하지 못하는것 같았다. 《K가 비밀이라면 자네는 왜 결과렬 $2^K \bmod N$, $3^K \bmod N$, $4^K \bmod N$이 한시각용암호화에 적합하지 않다고 생각하나?》

   《이보게, 와트손, K가 비밀이라고 해도 부분적이기는 하지만 그것을 예

견할수 없네. 자네는 암호분석자가 F와 렬의 일반적속성을 안다고 가정한다고 했지. 그럼 그가 출구렬의 짧은 토막을 얻어 낸다고 가정해 보라구. 암호체계에서는 이런 가정이 충분히 있을수 있는것으로 여기고 있네. 그리고 이 출구렬에 대하여 첫번째 두 원소만 알면 그것이 비록 전부는 아니라고 하여도 렬의 다음요소들을 상당히 많이 예견할수 있게 되며 결과 이 렬은 암호학적으로 강도가 높다고 볼수 없네. 또 좀 더 긴 토막을 알아 내려면 그는 다음요소들을 훨씬 더 많이 예견할수 있게 되지. 보라구. 렬의 일반적속성과 첫 두 원소2^K mod N과 3^K mod N만 알면 자네는 다음요소들을 쉽게 계산할수 있네…. 》

9. RSA는 문제 1의 행렬 M1, M2 및 M3에 의해 어떻게 표현될수 있는가?

10. 다음과 같은 방식을 고찰하자.
   1) 홀수 E를 취한다.
   2) 두 씨수 P와 Q를 취한다. 여기서 $(P-1)(Q-1)-1$은 E에 의해 완전히 나누어 진다.
   3) P와 Q를 곱하여 N을 구한다.
   4) $D=\dfrac{(P-1)(Q-1)(E-1)+1}{E}$을 계산한다.

   이 방식이 RSA와 동등한가? 왜 그런지 아닌지를 설명하시오.

11. B가 통보문을 암호화하여 A에게 보내는 다음과 같은 방식을 고찰하자.
   1) A는 $(P-1)$과 $(Q-1)$이 서로 소인 두개의 큰 씨수 P와 Q를 택한다.
   2) A는 자기의 공개열쇠로서 $N=PQ$를 공개한다.
   3) A는 $PP'\equiv 1 \pmod{Q-1}$이고 $QQ'\equiv 1 \pmod{P-1}$인 P'와 Q'를 계산한다.
   4) B는 M을 $C=M^N \pmod{N}$으로 암호화한다.
   5) A는 $M\equiv C^{P'} \pmod{Q}$와 $M\equiv C^{Q'} \pmod{P}$를 풀어 M을 구한다.

   ㄱ)이 방식이 어떻게 동작하는가를 설명하시오.
   ㄴ)이것이 RSA와 어떻게 차이나는가?
   ㄷ)이 방식에 비하여 RSA의 우점은 무엇인가?
   ㄹ)문제 1의 행렬 M1, M2 및 M3에 의해 어떻게 표현할수 있는가?

12. 《와트손, 이건 아주 흥미 있는 사건일세.》 홈스가 말하였다. 《한 총각이 처녀를 사랑하고 처녀 역시 총각을 사랑하거든. 그런데 처녀의 아버지가 별난 작자여서 사위감은 반드시 자기회사의 콤퓨터망에서 그가 리용하는데 알맞는 공개열쇠암호체계를 위한 간단하면서도 안전성이 담보된 규약을 설계해 내야 한다고 우기고 있다네. 그래서 그 젊은이는 량자간의, 실례로 사용자 B와 그에게 통보문 M을 보내려고 하는 사용자 A사이의 통신을 보장하기 위한 다음과 같은 규약을 작성했네(교환되는 통보문의 형식은 보내는 사람의 이름, 본문, 받는 사람의 이름으로 이루어 짐).

   1) A가 (A, E_{KU_b}M, A, B)를 B에게 보낸다.
   2) B가 (B, E_{KU_a}M, B, A)를 A에게 보내서 받았다는것을 인정한다.

   규약이 정말 간단하다는것은 자네도 알수 있을거야. 그런데 처녀의 아버지는 총각이 간단한 규약을 만들라는 자기의 요구를 만족시키지 못했다고 주장하고 있네. 그 리유인즉 그 규약에 군더더기가 있으며 다음과 같이 간략할수 있다는거네.

   1) A가 (A, E_{KU_b}M, B)를 B에게 보낸다.

2) B가 (B, E_{KU_a}[M], A)를 A에게 보내서 받았다는것을 인정한다.

이런 리유로 처녀의 아버지는 자기 딸이 그 총각에게 시집가는것을 허락하지 않았고 결국 두 사람은 불행하게 되였지. 그 총각이 나에게 도움을 청하러 금방 여기에 왔댔네.》《그래, 자네가 그를 꽤 도와 줄수 있을가?》 분명 와트손은 그 불쌍한 젊은이가 자기의 사랑을 잃어 버릴것같아 걱정하고 있었다.

《내 생각에는 도와 줄수 있을것 같네. 와트손, 자네도 알겠지만 군더더기가 있으므로 하여 때로는 규약의 보안을 보장하는데 유리할수도 있지. 처녀의 아버지가 간략한것으로 하여 오히려 그 새로운 규약이 원래의 규약은 이겨낼수 있던 공격을 받아 피해를 볼수 있게 되버렸네.》

홈스는 깊은 생각에 잠겨 말하였다.《그래, 와트손. 정말 그래. 보라구! 적이 바라는것은 바로 망사용자들중의 하나가 되여 A와 B사이에 오가는 통보문을 가로 채는것이지. 망의 한 사용자가 됨으로써 그는 자기자신의 공개암호열쇠를 가지고 자기의 통신을 A나 B에게 보낼수도 있고 그들의것을 받을수도 있게 되지. 축소 간략화된 규약의 도움으로 그가 다음의 절차를 리용하고 있는 B에게 사용자 A가 이전에 보낸 통보문 M을 그때에는 얻을수 있게 되지. …》

13. 여기에는 다른 한가지 고속제곱연산알고리듬이 있다. 이것이 그림 6-7의것과 동등하다는것을 보여 주시오.
   1) d←1;T←a;E←b
   2) **if** odd(e) **then** d←d×T
   3) E←⌊E/2⌋
   4) T←T×T
   5) **if** E>0 **then goto** 2
   6) **output** d
14. 공통씨수 q=11이고 원시뿌리 α=2인 디피-헬만을 고찰하자.
   ㄱ)사용자 A가 공개열쇠 Y_A=9를 가지면 A의 비밀열쇠 X_A는 무엇인가?
   ㄴ)사용자 B가 공개열쇠 Y_B=3을 가지면 공유된 비밀열쇠 K는 무엇인가?
15. 《그러나》하고 와트손은 말하였다.《당신의 의뢰자들은 자기들의 망에서 디피-헬만열쇠교환규약을 리용하네. 그것은 리산로그에 기초하고 있는데 이것이 힘든 문제의 하나로 알려 져 있지?》

   《옳네, 와트손.》홈스는 머리를 끄떡이였다.《적당한 파라메터선택에 대한 리산로그문제는 정말 어렵지. 나의 의뢰자들이 그것을 알고 있고 바로 그래서 그들은 이 열쇠배포방법을 택한거야. 불행하게도 그들의 안전담당전문가들은 능동적인 적이 종종 피동적인 적보다 더 성공적이라는것을 깨닫지 못했네. 적은 또한 자기가 적당한 기회에 리산로그문제를 풀수 없다는것과 그래서 그 어떤 다른것을 시도해야 한다는것도 알고 있지. 그리고 나는 모리어티(Moriarty)자신이 나의 의뢰자들의 통신에 흥미를 가지고 있다는것을 확신하며 그래서 나는 그들의 망에 대한 어떤 종류의 적극적인 공격을 예상해야 하거든. 와트손, 모리어티는 절대로 피동적으로 가만히 있을 사람이 아니야.》와트손이 놀라서 물었다.《홈스, 자넨 모리어티가 디피-헬만열쇠교환방식을 파괴할 방법을 찾아 낼수 있으리라고 생각하나?》

《아, 그건 그리 어려운게 아니거든.》 홈스는 미소를 지었다.

《모리어티가 바라는것은 바로 모든 통신을 가로 챌뿐아니라 파괴해 버릴 수 있는 통신경로안에 자기의 위치를 잡자는거야. 나는 모리어티가 그것을 충분히 해낼수 있는 능력이 있다고 보네. 이제 그러한 위치에 꽉 …》

16. 타원곡선 $E_{11}(1,6)$을 고찰하자. 즉 mod $p=11$에 관하여 $y^2=x^3+x+6$으로 곡선이 정의된다. $E_{11}(1,6)$의 모든 점들을 구하시오. 암시: x의 모든 값에 대하여 등식의 오른변을 구하는것부터 시작하시오.

17. $E_{11}(1,6)$에 대하여 점 $G=(2,7)$을 고찰하자. $2G$부터 $13G$까지 G의 배수들을 구하시오.

18. 이 문제는 6.5에서 강조된 방식을 리용하여 타원곡선암호화/복호화를 진행한다. 암호체계의 파라메터는 $E_{11}(1,6)$과 $G=(2,7)$이다. B의 비밀열쇠는 $n_B=7$이다.
   ㄱ) B의 공개열쇠 P_B를 구하시오.
   ㄴ) A가 통보문 $P_m=(10,9)$를 암호화하고 란수 $k=3$을 취한다. 암호문 C_m을 구하시오.
   ㄷ) B가 C_m을 P_m으로 회복하는 계산과정을 보여 주시오.

19. 1985년에 티 엘가말(T.ElGamal)은 디피-헬만기술과 밀접한 련관을 가지는 리산로그에 기초한 공개열쇠방식을 발표하였다. 디피-헬만에서처럼 엘가말방식의 대역적요소는 씨수 q와 q의 원시뿌리 α이다. 사용자 A는 비밀열쇠 X_A를 취하고 디피-헬만에서처럼 공개열쇠 Y_A를 계산한다. 사용자 A는 다음과 같이 사용자 B에게 보낼 평문 $M<q$를 암호화한다.
   1) $1 \le k \le q-1$인 옹근수 k를 택한다.
   2) $K=(Y_B)^k \pmod q$를 계산한다.
   3) M을 옹근수쌍 (C_1, C_2)로 암호화한다. 여기서 $C_1=\alpha^k \pmod q$, $C_2=KM \pmod q$이다.

   사용자 B는 다음과 같이 하여 평문 M을 얻는다.
   1) $K=(C_1)^{X_B} \pmod q$를 계산한다.
   2) $M=(C_2K^{-1}) \pmod q$를 계산한다.

   체계의 동작을 보여 주라. 즉 복호화처리가 평문을 얻는다는것을 보여 주시오.

20. 공통씨수 $q=71$이고 원시뿌리 $\alpha=7$인 엘가말방식을 고찰하자.
   ㄱ) B의 공개열쇠가 $Y_B=3$이고 A가 우연옹근수 $k=2$를 택하면 $M=30$의 암호문은 무엇인가?
   ㄴ) 이제 A가 k의 다른 값을 취하여 $M=30$의 암호문이 $C=(59, C_2)$이라면 옹근수 C_2은 무엇인가?

21. 부록 6의 알고리듬 P1을 개선한다.
   ㄱ) $2n$번의 곱하기와 $n+1$번의 더하기를 요구하는 알고리듬을 개발하시오.
   암시: $x^{i+1}=x^i \times x$.
   ㄴ) 오직 $n+1$번의 곱하기와 더하기만을 요구하는 알고리듬을 개발하시오.
   암시: $P(x)=a_0+x \times q(x)$. 여기서 $q(x)$는 $(n-1)$차다항식이다.

부록 6: 알고리듬의 복잡성

암호알고리듬을 분석하는데서 기본문제는 주어 진 공격을 취하는 시간량이다. 전형적으로 가장 효과적인 공격알고리듬을 구하였다는것을 믿을수 없다. 사람들이 그렇게 말할수 있는 가장 중요한것은 특별한 알고리듬에 대하여 공격에 대한 효과준위가 특별한 크기순서이라는것이다. 그때에야 특별한 알고리듬의 보안준위를 결정하는 현재 또는 예언된 처리의 속도에 대한 크기순서를 비교할수 있다.

알고리듬의 효과성에 대한 공통적인 척도는 그의 시간복잡성이다. 모든 n과 길이 n인 모든 입력에 대하여 알고리듬의 실행이 기껏해야 f(n)개 단계를 취한다면 그 알고리듬의 시간복잡성은 f(n)이라고 정의한다. 이리하여 주어 진 크기의 입력과 처리속도에 대하여 시간복잡성은 실행시간에 관한 웃한계이다.

여기에는 몇가지 애매성이 있다. 우선 단계의 정의가 정확하지 않다. 단계는 튜링기계, 하나의 처리기계명령, 하나의 고준위언어명령 등의 단순한 조작이다. 그러나 단계에 대한 이런 여러가지 정의는 모두 곱하기상수와 관련되여 있다. n의 아주 큰 값에 대하여 이런 상수들은 중요하지 않다. 중요한것은 관련되는 실행시간이 얼마나 빨리 성장하는가이다. 실례로 RSA에 대하여 50자리($n=10^{50}$) 또는 100자리($n=10^{100}$)의 열쇠를 리용하는가 아닌가에 대해 고찰한다면 매 열쇠의 크기를 파괴하는데 정확히 얼마만한 시간이 걸리는가를 아는것은 필요 없다(또는 실제적으로 불가능하다). 오히려 효과준위를 보여 주는 원형그림부분에만 관심을 가지게 되여 극적으로 관계되는 효과가 큰 열쇠크기에 대하여 얼마나 요구되는가를 알고 싶어 한다.

다음으로 제기되는 문제는 일반적으로 말하여 f(n)에 대한 정확한 식을 끌어 낼수 없다는것이다. 오직 그것을 근사시킬수만 있다. 그러나 다시 초기에 n이 커지는데 따라 f(n)의 변화비률에 관심을 가지게 된다.

알고리듬의 시간복잡성을 특징 짓는 개념으로서 큰 기호 O를 사용하는 표준적인 수학적표시가 있다. 그 정의는 다음과 같다. f(n)=O(g(n))이기 위해서는 두개의 수 a와 M이 있어서

$$|f(n)| \leq a \times |g(n)|, \qquad n \geq M \qquad (6\text{-}2)$$

일것이 필요하고 충분하다.

실례를 통하여 이 개념을 명백히 하자.

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

형태의 일반다항식을 평가하려고 한다고 가정하자. 다음의 알고리듬은 [POHL81]에서 나온것이다.

```
algorithm P1;
        n,i,j:integer; x,polyval:real;
        a,s:array[0..100] of real;
        begin
                read(x,n);
                for i:=0 upto n do
```

```
        begin
                s[i]:=1; read(a[i]);
                for j:=1 upto i do s[i]:=x×s[i];
                s[i]:=a[i]×s[i]
        end;
        polyval:=0;
        for i:=0 upto n do polyval:=polyval+s[i];
        write('value at', x, 'is', polyval)
end.
```

이 알고리듬에서 매 부분식들은 따로따로 평가된다. 매 s[i]는 (i+1)번의 곱하기 즉 s[i]를 계산하는데 i번의 곱하기와 a[i]로 곱하기 한번을 요구한다. n개 항모두를 계산하는데는

$$\sum_{i=0}^{n}(i+1) = \frac{(n+2)(n+1)}{2}$$

번의 곱하기를 요구한다. 또한 아주 많은 회수의 곱하기에 비하면 무시할수 있는 (n+1)번의 더하기가 있다. 이리하여 이 알고리듬의 시간복잡성은 f(n) = (n+2)(n+1)/2이다. 이제 $f(n)=O(n^2)$임을 보자. 식 6-2의 정의로부터 a=1과 M=4에 대하여 이 식은 g(n) $=n^2$로 성립한다는것을 보여 주자. 이것을 n에 관한 귀납법으로 증명한다. n=4일 때 성립한다. 왜냐하면 $(4+2)(4+1)/2=15 < 4^2=16$이기때문이다. 이제 이것이 k까지의 모든 n값에 대하여 성립한다는것을 가정하자. 이때 $n=k+1$라고 하면

$$\begin{aligned}\frac{(n+2)(n+1)}{2} &= \frac{(k+3)(k+2)}{2}\\ &= \frac{(k+2)(k+1)}{2}+k+2\\ &\le k^2+k+2\\ &\le k^2+2k+2=(k+1)^2=n^2\end{aligned}$$

그러므로 결과는 $n=k+1$일 때 참이다.

일반적으로 큰 기호 O개념은 가장 빨리 성장한다는 술어를 리용하게 한다. 실례로

1. $O(ax^7+3x^3+\sin(x))=O(ax^7)=O(x^7)$
2. $O(e^n+an^{10})=O(e^n)$
3. $O(n!+n^{50})=O(n!)$

큰 기호 O개념에 대하여 보다 매력적인것들이 많이 있다. 관심 있는 독자들을 위하여 한가지 지적해 줄 도서는 [GRAH94]이다.

크기 n인 입력을 가진 알고리듬은

- 실행시간이 O(n)이면 **선형**이다.
- 실행시간이 어떤 상수 t에 대하여 O(n^t)이면 **다항식**이다.
- 실행시간이 어떤 상수 t와 다항식 h(n)에 대하여 O($t^{h(n)}$)이면 **지수적**이다

고 말한다.

일반적으로 다항식시간내에 풀수 있는 문제를 가능한것으로 고찰하며 다항식시간보다 더 나쁜것 특히 지수함수적시간을 불가능한것으로 고찰한다. 그러나 이런 술어들에 주의하여야 한다. 우선 입력의 크기가 충분히 작다면 아주 복잡한 알고리듬조차도 가능한것으로 된다. 실례로 단위시간당 10^{12}개연산을 실행할수 있는 체계가 있다고 가정하자. 표 6-6은 여러가지 복잡성에 관한 알고리듬을 한시간단위로 조종할수 있는 입력의 크기를 보여 준다. 지수함수적시간 또는 차례곱시간의 알고리듬에 대하여 오직 작은 입력만을 달성할수 있다.

표 6-6. 여러가지 복잡성수준의 효과수준

복잡성	크기	연산
$\log_2 n$	$2^{10^{12}}=10^{3\times10^{11}}$	10^{12}
n	10^{12}	10^{12}
n^2	10^6	10^{12}
n^6	10^2	10^{12}
2^n	28	10^{12}
$n!$	15	10^{12}

다음으로 주의해야 할 문제는 입력을 특징 짓는 방법이다. 실례로 암호알고리듬의 암호분석의 복잡성을 열쇠의 길이나 가능한 열쇠의 개수에 의하여 동등하게 잘 특징지을수 있다. DES에 대하여 실례로 가능한 열쇠의 개수는 2^{56}이며 열쇠의 길이는 56bit이다. 하나의 암호화를《단계》로, 가능한 열쇠의 개수를 $N=2^n$으로 고찰한다면 알고리듬의 시간복잡성은 n의 개수에는 선형O(N)이지만 열쇠의 길이에는 지수함수적 O(2^n)이다.

제 7 장. 수론초보

공개열쇠알고리듬에서는 일련의 수론적개념들이 본질적이다. 이 장에서는 다른 장들에서 참고로 쓰이는 개념들을 개괄한다. 이런 개념을 알고 있는 독자들은 이 장을 뛰여넘을수도 있다.

수론의 개념들과 기교들은 아주 추상적이므로 실례없이 직접 습득하는것은 힘들다 [RUB97b]. 그러므로 이 장에서는 실례들을 많이 준다.

7.1 씨수와 서로 소

수론에서 중심적개념은 씨수이다. 그러므로 이 제목에서는 이 책에서 필요로 하는 씨수에 관한 내용들을 개괄한다. 특별히 강조하지 않는 한 부아닌 옹근수만을 취급한다.

약수

어떤 m에 대하여 $a=mb$이면 $b\neq0$은 a를 나눈다고 한다. 여기서 a와 b,m은 옹근수이다. 즉 나누기에 관하여 나머지가 없으면 b는 a를 나눈다. b가 a를 나눈다는것을 $b|a$로 표시한다. 또한 $b|a$이면 b를 a의 약수라고 한다.

24 의 약수는 1, 2, 3, 4, 6, 8, 12, 24 이다.

다음의 관계가 성립한다.

- $a|1$ 이면 $a=\pm 1$ 이다.
- $a|b$ 이고 $b|a$ 이면 $a=\pm b$ 이다.
- 임의의 $b\neq0$ 은 0 을 나눈다.
- $b|g$ 이고 $b|h$ 이면 임의의 옹근수 m 과 n 에 대하여 $b|(mg+nh)$이다.

마지막관계를 고찰하자.

$b|g$ 이면 g 는 어떤 옹근수 g_1 에 대하여 $g=b\times g_1$ 이고
$b|h$ 이면 h 는 어떤 옹근수 h_1 에 대하여 $h=b\times h_1$

이므로

$$mg+nh=mbg_1+nbh_1=b\times(mg_1+nh_1)$$

이다. 따라서 b 는 $mg+nh$ 를 나눈다.

$b = 7$; $g = 14$; $h = 63$; $m = 3$; $n = 2$
$7|14$ 이고 $7|63$ 이다. $7|(3 \times 14 + 2 \times 63)$이라는것을 보자.
$(3 \times 14 + 2 \times 63) = 7(3 \times 2 + 2 \times 9)$로부터 분명히
$7|(7(3 \times 2 + 2 \times 9))$.

씨수

어떤 옹근수 $p>1$의 약수들이 오직 ± 1과 $\pm p$뿐이면 p를 씨수라고 한다. 씨수는 수론에서 그리고 이 장에서 론의되는 기법들에서 결정적역할을 한다.

표 7-1은 2000보다 작은 씨수들을 보여 준다.

임의의 옹근수 $a>1$을

$$a = p_1^{\alpha_1} p_2^{\alpha_2} \cdots \cdots p_t^{\alpha_t}$$

와 같이 씨수들의 적으로 유일하게 인수분해할수 있다. 여기서 $p_1<p_2<\cdots<p_t$ 이고 매 $\alpha_i>0$ 이다.

$91 = 7 \times 13$; $11011 = 7 \times 11^2 \times 13$

이 표식을 리용하면 다음과 같은 점에서 유리하다. P가 씨수모두의 모임이면 임의의 정의옹근수를 다음과 같은 형식으로 유일하게 표시할수 있다.

$$a = \prod_{p} p^{a_p}, \qquad \text{여기서 } a_p \geqq 0$$

오른변은 가능한 씨수 p모두우에서의 적이며 임의의 a의 매개 값에 대하여 제곱지수 a_p의 대부분은 0이다.

$3600 = 2^4 \times 3^2 \times 5^2$

임의의 주어 진 정의옹근수의 값을 우의 공식화에서 령아닌 제곱지수모두를 간단히 렬거할수 있다.

옹근수 12는 $\{a_2=2, a_3=1\}$로 표현된다.
옹근수 18은 $\{a_2=1, a_3=2\}$로 표현된다.

두 수의 곱하기는 대응하는 제곱지수들의 더하기와 동등하다. 즉 모든 p에 대하여

$$k = mn \rightarrow k_p = m_p + n_p$$

$k = 12 \times 18 = 216$
$k_2 = 2 + 1 = 3$; $k_3 = 1 + 2 = 3$
$216 = 2^3 \times 3^3$

표 7-1. 2000 보다 작은 씨수

2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	61	67	71	73	79	83	89	97
101	103	107	109	113	127	131	137	139	149	151	157	163	167	173	179	181	191	193	197	199				
211	223	227	229	233	239	241	251	257	263	269	271	277	281	283	293									
307	311	313	317	331	337	347	349	353	359	367	373	379	383	389	397									
401	409	419	421	431	433	439	443	449	457	461	463	467	479	487	491	499								
503	509	521	523	541	547	557	563	569	571	577	587	593	599											
601	607	613	617	619	631	641	643	647	653	659	661	673	677	683	691									
701	709	719	727	733	739	743	751	757	761	769	773	787	797											
809	811	821	823	827	829	859	853	857	859	863	877	881	883	887										
907	911	919	929	937	941	947	953	967	971	977	983	991	997											
1009	1013	1019	1021	1031	1033	1039	1049	1051	1061	1063	1069	1087	1091	1093	1097									
1103	1109	1117	1123	1129	1151	1153	1163	1171	1181	1187	1193													
1201	1213	1217	1223	1229	1231	1237	1249	1259	1277	1279	1283	1289	1291	1297										
1301	1303	1307	1319	1321	1327	1361	1367	1373	1381	1399														
1409	1423	1427	1429	1433	1439	1447	1451	1453	1459	1471	1481	1483	1487	1489	1493	1499								
1511	1523	1531	1543	1549	1553	1559	1567	1571	1579	1583	1597													
1601	1607	1609	1613	1619	1621	1627	1637	1657	1663	1667	1669	1693	1697	1699										
1709	1721	1723	1733	1741	1747	1753	1759	1777	1783	1787	1789													
1801	1811	1823	1831	1847	1861	1867	1871	1873	1877	1879	1889													
1901	1907	1913	1931	1933	1949	1951	1973	1979	1987	1993	1997	1999												

이것은 $a|b$ 인 씨인수들에 관하여 무엇을 의미하는가? p^k 형태의 임의의 옹근수는 오직 $j \le k$ 인 p^j 형태의 옹근수로만 나누어 질수 있다. 이리하여 모든 p 에 대하여

$$a|b \rightarrow a_p \le b_p$$

이라는것을 알수 있다.

$a = 12$; $b=36$; $12|36$; $12 = 2^2 \times 3$; $36 = 2^2 \times 3^2$
$a_2 = 2 = b_2$
$a_3 = 1 \le 2 = b_3$

서로 소

두 수 a와 b의 최대공약수(greatest common divisor) $\gcd(a,b)$를 리용하자. 다음의 조건들을 만족시키면 정의옹근수 c를 a와 b의 최대공약수라고 한다.

1. c 는 a 와 b 의 약수이다.
2. a 와 b 의 임의의 약수는 c 의 약수이다.

동등한 정의는 다음과 같다.

$$\gcd(a,b) = \max[k,\ k|a \text{ 이고 } k|b]$$

최대공약수는 정수일것을 요구하므로

$$\gcd(a,b) = \gcd(a,-b) = \gcd(-a,b) = \gcd(-a,-b).$$

일반적으로 $\gcd(a,b) = \gcd(|a|,\ |b|)$.

$\gcd(60,24) = \gcd(60,-24) = 12$

또한 모든 령아닌 옹근수들은 0 을 나누므로 $\gcd(a,0)=|a|$.

모든 옹근수들을 씨수들의 적으로 표시하면 두 정의옹근수의 최대공약수를 결정하는 것은 아주 쉽다.

$$300 = 2^2 \times 3^1 \times 5^2$$
$$18 = 2^1 \times 3^2$$
$$\gcd(18,300) = 2^1 \times 3^1 \times 5^0 = 6$$

일반적으로 모든 p에 대하여

$$k = \gcd(a,\ b) \rightarrow k_p = \min(a_p,\ b_p)$$

큰 수의 씨인수를 결정하는것은 쉬운 일이 아니므로 우의 관계식을 최대공약수를 계산하는데 직접 리용할수 없다. 7.5에서 이에 대한 론의를 한다.

두 옹근수 a와 b가 공통의 씨인수를 가지지 않는다면 즉 공통의 인수가 오직 1뿐이면 a와 b는 서로 소이다고 한다. 이것은 $\gcd(a,b)=1$이면 a와 b는 서로 소이다고 하는 것과 동등하다.

8의 약수는 1, 2, 4, 8이고 15의 약수는 1, 3, 5,15 이므로 8과 15는 서로 소이다.

7.2 Mod산수

임의의 정의옹근수 n과 임의의 옹근수 a가 주어 졌을 때 a를 n으로 나누면 다음의 식이 성립하는 상 q와 나머지 r를 얻는다.

$$a = qn + r \qquad 0 \le r < n; \; q = \lfloor a/n \rfloor$$

여기서 $\lfloor x \rfloor$는 x보다 크지 않은 최대옹근수이다.

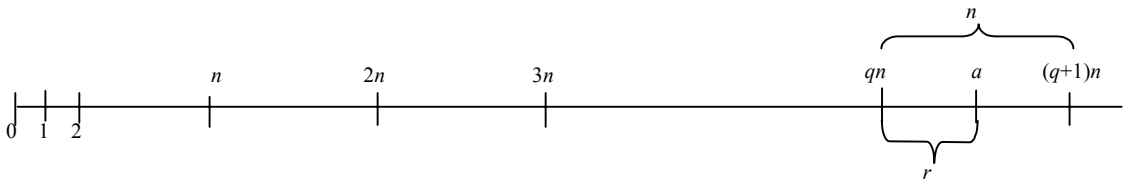



그림 7-1. 식 $a = qn + r; \; 0 \le r < n$

그림 7-1에서는 a와 정의 n이 주어 졌을 때 우의 식을 만족시키는 q와 r를 구하는것은 항상 가능하다는것을 보여 준다. 수직선우에 옹근수를 표시하자. 즉 a를 수직선우에 표시하자. 0부터 시작하여 $qn \le a$이고 $(q+1)n > a$인 qn까지 n, $2n$, …, 등으로 표시해나간다. qn으로부터 a까지의 거리가 r이므로 q와 r의 유일한 값을 구하였다. 나머지 r를 때때로 잉여라고도 한다.

$a = 11$; $n = 7$; $11 = 1 \times 7 + 4$; $r = 4$
$a = -11$; $n = 7$; $-11 = (-2) \times 7 + 3$; $r = 3$

a가 옹근수이고 n이 정의옹근수이면 $a \bmod n$을 a를 n으로 나누었을 때의 나머지로 정의한다. 이리하여 임의의 옹근수 a에 대하여 항상 다음과 같이 쓸수 있다.

$$a = \lfloor a/n \rfloor \times n + (a \bmod n)$$

11 mod 7 = 4; -11 mod 7 = 3

두 옹근수 a와 b에 대하여 $(a \bmod n)=(b \bmod n)$이면 a와 b는 모드 n에 관하여 **합동**이라고 한다. 이것을 $a \equiv b \bmod n$으로 표시한다.

$$73 \equiv 4 \bmod 23; \quad 21 \equiv -9 \bmod 10$$

$a \equiv 0 \bmod n$이면 $n|a$이라는것을 강조한다.
모드연산자는 다음의 성질들을 만족시킨다.

1. $n|(a-b)$이면 $a \equiv b \bmod n$
2. $(a \bmod n)=(b \bmod n)$이면 $a \equiv b \bmod n$
3. $a \equiv b \bmod n$이면 $b \equiv a \bmod n$
4. $a \equiv b \bmod n$이고 $b \equiv c \bmod n$이면 $a \equiv c \bmod n$

첫번째 성질을 고찰하자. $n|(a-b)$이면 어떤 k에 대하여 $(a-b)=kn$이다. 따라서 $a=b+kn$으로 쓸수 있다. 그러므로

$(a \bmod n)=(b+kn$을 n으로 나누었을 때의 나머지)
$=(b$를 n으로 나누었을 때의 나머지)
$=(b \bmod n)$

$23-8=15=5\times 3$	이므로	$23 \equiv 8 \pmod 5$
$-11-5=-16=8\times(-2)$	이므로	$-11 \equiv 5 \pmod 8$
$81-0=81=27\times 3$	이므로	$81 \equiv 0 \pmod{27}$

나머지성질들도 쉽게 증명된다.

Mod 산수연산

정의(그림 7-1)로부터 (mod n)연산자들은 모든 옹근수를 옹근수모임 $\{0, 1, 2, \cdots (n-1)\}$안에로 넘긴다. 이때 다음과 같은 문제가 제기된다. 이 모임안에서 산법연산을 진행할수 있겠는가? 그것을 다음과 같이 할수 있다. 그 기법이 바로 **Mod산수**로 알려 져 있다.
Mod 산수는 다음의 성질들을 만족시킨다.

1. $[(a \bmod n)+(b \bmod n)] \bmod n=(a+b) \bmod n$
2. $[(a \bmod n)-(b \bmod n)] \bmod n=(a-b) \bmod n$
3. $[(a \bmod n)\times(b \bmod n)] \bmod n=(a\times b) \bmod n$

첫째 성질을 고찰하자. $(a \bmod n)=r_a$이고 $(b \bmod n)=r_b$라고 하자. 이때 어떤 옹근수 j에 대하여 $a=r_a+jn$이고 어떤 옹근수 k에 대하여 $b=r_b+kn$이다. 그러므로

$$\begin{aligned}(a+b) \bmod n &= (r_a+jn+r_b+kn) \bmod n \\ &= [r_a+r_b+(k+j)n] \bmod n \\ &= (r_a+r_b) \bmod n \\ &= [(a \bmod n)+(b \bmod n)] \bmod n\end{aligned}$$

나머지성질들도 쉽게 증명된다. 3개 성질들에 대한 실례가 있다.

11 mod 8 = 3; 15 mod 8 = 7
[(11 mod 8)+(15 mod 8)] mod 8 = 10 mod 8 = 2
(11 + 15) mod 8 = 26 mod 8 = 2
[(11 mod 8)−(15 mod 8)] mod 8 =-4 mod 8 = 4
(11-15) mod 8 = -4 mod 8 = 4
[(11 mod 8)×(15 mod 8)] mod 8 = 21 mod 8 = 5
(11 × 15) mod 8 = 165 mod 8 = 5

제곱연산을 보통의 산법연산에서처럼 곱하기산법연산의 반복으로 진행할수 있다(7.7에서 제곱연산을 구체적으로 론의한다).

11^7 mod 13을 다음과 같이 구할수 있다.
$11^2 = 121 \equiv 4 \mod 13$
$11^4 \equiv 4^2 \equiv 3 \mod 13$
$11^7 \equiv 11 \times 4 \times 3 \equiv 132 \equiv 2 \mod 13$

이리하여 더하기와 덜기,곱하기를 가지는 보통의 산법규칙들이 Mod 산수에 대해서도 성립한다.

표 7-2는 mod 8에 관한 더하기와 곱하기를 보여 준다. 더하기를 보면 결과는 간단하게 행렬에서 4각형안에 있다. 또한 보통의 더하기에서처럼 Mod산수의 매수에 대하여 더하기반대수 즉 부수가 있다.
이 경우에 수 x의 부수는 $x+y=0$ mod 8인 수 y이다. 왼쪽렬에서 어떤 수의 더하기반대수를 구하려면 우선 행렬에서 대응하는 행을 조사하여 값 0을 찾는다. 그다음에는 그 행의 꼭대기수가 더하기반대수이다. 이리하여 2+6=0 mod 8. 류사하게 곱하기표에서 입구들도 간단화된다. 보통의 산법에서는 매수에 대하여 곱하기역수가 있다. 모드 8에 관한 Mod산수에서는 x의 곱하기역수는 $x \times y \equiv 1$ mod 8인 수 y이다. 이제 곱하기표에서 어떤 수의 곱하기역수를 구하려면 우선 그 수가 있는 행을 조사하여 값 1을 구한다. 그다음 그 렬의 꼭대기수가 곱하기역수이다. 이리하여 3×3=1 mod 8이다. mod 8에 관한 모든 수들이 곱하기역수를 가지는것은 아니라는것을 강조한다. 이것을 후에 론의한다.

Mod 산수의 성질

모임 Z_n을 n보다 작은 부아닌 옹근수들의 모임으로 정의한다. 즉

$$Z_n=\{0, 1, 2, ..., (n-1)\}$$

이것을 모드 n에 관한 나머지들의 모임이라고 한다. 이 모임안에서 Mod산수를 진행하면 다음의 성질들이 Z_n의 옹근수들에 대하여 성립한다.

표 7-2. 모드 8에 관한 산법

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

ㄱ) 모드 8에 관한 더하기

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

ㄴ) 모드 8에 관한 곱하기

Mod 산수는 보통의 산법과 차이나는 하나의 특성이 있다. 첫째로 보통의 산법에서처럼 다음과 같은것을 쓸수 있다.

$(a + b) \equiv (a + c) \bmod n$ **이면** $b \equiv c \bmod n$ (7-1)

$(5 + 23) \equiv (5 + 7) \bmod 8; \quad 23 \equiv 7 \bmod 8$

식 7-1은 더하기반대수의 존재성에 모순되지 않는다. 식 7-1의 량변에 a의 더하기반대수를 더하면

성질	**식**
가환성	$(w + x) \bmod n = (x + w) \bmod n$
	$(w \times x) \bmod n = (x \times w) \bmod n$
결합성	$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$
	$[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
배송성	$[w \times (x + y)] \bmod n = [(w + x) + (w + y)] \bmod n$
단위성	$(0 + w) \bmod n = w \bmod n$
	$(1 \times w) \bmod n = w \bmod n$
더하기반대수$(-w)$	매 $w \in Z_n$에 대하여 $w + z \equiv 0 \bmod n$인 z가 있다.

$$((-a) + a + b) \equiv ((-a) + a + c) \mod n$$
$$b \equiv c \mod n$$

그러나 다음의 명제는 일정한 조건하에서만 성립한다.

a가 n과 서로 소이라는 조건하에서 $(a \times b) \equiv (a \times c)$**이면** $b \equiv c \mod n$ (7-2)

우의 조건이 성립하지 않는 실례
$6 \times 3 = 18 \equiv 2 \mod 8$
$6 \times 7 = 42 \equiv 2 \mod 8$
그러나 3은 mod 8에 관하여 7과 합동이 아니다.

이런 현상이 생기는 원인은 임의의 일반적인 mod n에 대하여 인수 a를 수 0부터 $(n-1)$사이에 실제로 적용할 때 a와 n이 공통인수를 가지면 나머지들의 완비모임이 얻어지지 않기때문이다.

$a = 6$ 이고 $n = 8$ 일 때

Z_8	0	1	2	3	4	5	6	7
6 으로 곱하기	0	6	12	18	24	30	36	42
나머지	0	6	4	2	0	6	4	2

6으로 곱할 때 나머지들이 완비모임으로 되지 않기때문에 Z_8의 여러수들이 같은 나머지에로 넘어 간다. 특히 6×0 mod 8=6×4 mod 8이며 6×1 mod 8=6×5 mod 8 등등이다. 이것은 다 대 일 넘기기이므로 곱하기연산에 관하여 유일한 역수가 없다.
그러나 a=5이고 n=8을 취하면

Z_8	0	1	2	3	4	5	6	7
5 . .곱하기	0	5	10	15	20	25	30	35
ㅏ머지	0	5	2	7	4	1	6	3

나머지행은 Z_8의 모든 수들을 서로 다른 순서로 포함한다.

최종적으로 p가 씨수이면 Z_p의 모든 원소들은 p와 서로 소이다. 이것은 우에서 렬거된 다음과 같은 하나의 성질을 가질수 있게 한다.

곱하기역수 (w^{-1})	매 $w \in Z_p$에 대하여 $w \times z \equiv 1 \mod p$인 z가 있다.

w는 p와 서로 소이므로 Z_p의 모든 원소들을 w로 곱하면 그 결과의 나머지들은 Z_p의 치환된 원소모두를 포괄한다. 이리하여 나머지들중의 적어도 하나는 값 1을 가진다. 그러므로 w로 곱할 때 Z_p의 어떤 수가 있어서 나머지 1로 된다. 이 수가 바로 w의 곱하기역수이다. 이것을 w^{-1}로 표시한다. 이리하여 식 7-2는 곱하기역수의 존재성에 모순되지

않는다. 식 7-2의 량변을 a의 곱하기역수로 곱하면

$$((a^{-1})\times a\times b)\equiv((a^{-1})\times a\times c)\bmod n$$
$$b\equiv c \bmod n$$

마지막주의점: 일부 옹근수들은 모드들이 씨수가 아닐지라도 곱하기역수를 가진다. 가령 $\gcd(a,n)=1$이면 $a\times b\equiv 1 \bmod n$인 Z_n의 b를 구할수 있다. 그 리유는 우에서와 같다. a가 n과 서로 소이므로 Z_n의 모든 원소들을 a로 곱하면 그 결과의 나머지들은 Z_n의 치환된 원소모두를 포괄한다. 그러므로 Z_n의 어떤 수 b가 있어서 $a\times b\equiv 1 \bmod n$이다.

표 7-3은 이 절의 개념들을 보여 주는 실례이다.

표 7-3. 모드 7에 관한 산법

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

ㄱ) 모드 7에 관한 더하기

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

ㄴ) 모드 7에 관한 곱하기

w	$-w$	w^{-1}
0	0	—
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

ㄷ) 모드 7에 관한 더하기반대수와 곱하기역수

7.3 페르마정리와 오일레르정리

공개열쇠암호에서 중요한 역할을 하는 두 정리는 페르마정리와 오일레르정리이다.

페르마정리

페르마정리는 다음과 같다. p가 씨수이고 a가 p로 나누어 지지 않는 정의옹근수이면

$$a^{p-1} \equiv 1 \bmod p \tag{7-3}$$

증명: 앞절의 론의로부터 Z_p의 원소모두를 mod p에 관하여 a로 곱하면 그 결과는 일정한 순서로 Z_p의 원소들을 이룬다. 그리고 $a \times 0 \equiv 0 \bmod p$이다. 그러므로 $(p-1)$개의 수 $\{a \bmod p,\ 2a \bmod p,\ \cdots\ (p-1)a \bmod p\}$들은 일정한 순서로 수 $\{1, 2, \cdots, (p-1)\}$로 된다. 이 수들을 다음과 같이 곱하자.

$$\begin{aligned} a \times 2a \times \cdots \times ((p-1)a) &\equiv [(a \bmod p) \times (2a \bmod p) \times \cdots \times ((p-1)a \bmod p)] \bmod p \\ &\equiv (p-1)! \bmod p \end{aligned}$$

그러나

$$a \times 2a \times \cdots \times [(p-1)a] = (p-1)!\, a^{p-1}$$

그러므로

$$(p-1)!\, a^{p-1} \equiv (p-1)! \bmod p$$

$(p-1)!$은 p와 서로 소이므로 그것을 제거할수 있다(식 7-2를 보시오). 이로부터 식 7-3이 나온다.

$a = 7,\ p = 19$
$7^2 = 49 \equiv 11 \bmod 19$
$7^4 \equiv 121 \equiv 7 \bmod 19$
$7^8 \equiv 49 \equiv 11 \bmod 19$
$7^{16} \equiv 121 \equiv 7 \bmod 19$
$a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1 \bmod 19$

이 정리에 대한 다른 하나의 형식이 또한 유용하다. 즉 p가 씨수이고 a가 정의옹근수이면

$$a^p \equiv a \bmod p \tag{7-4}$$

$p = 5,\ a = 3,\ 3^5 = 243 \equiv 3 \bmod 5$
$p = 5,\ a = 10,\ 10^5 = 100000 = 10 \bmod 5 \equiv 0 \bmod 5$

오일레르함수

오일레르정리를 제기하기전에 수론에서의 중요한 오일레르함수 $\phi(n)$을 도입할 필요가 있다. 여기서 $\phi(n)$은 n보다 작으며 n과 서로 소인 정의옹근수들의 수이다.

표 7-4는 $\phi(n)$의 첫 30개 값들을 렬거한다. 값 $\phi(1)$은 의미가 없지만 값 1로 정의한다.

표 7-4. 오일레르함수 $\phi(n)$의 몇가지 값

n	$\phi(n)$	n	$\phi(n)$	n	$\phi(n)$
1	1	11	10	21	12
2	1	12	4	22	10
3	2	13	12	23	22
4	2	14	6	24	8
5	4	15	8	25	20
6	2	16	8	26	12
7	6	17	16	27	18
8	4	18	6	28	12
9	6	19	18	29	28
10	4	20	8	30	8

씨수 p에 대하여 분명히

$$\phi(p) = p - 1$$

이다. 이제 두 씨수 p와 q가 있다고 가정하자. 이때 $n=pq$에 대하여

$$\phi(n) = \phi(pq) = \phi(p) \times \phi(q) = (p-1) \times (q-1)$$

이것을 보기 위하여 Z_n의 나머지들의 모임이 $\{0, 1, 2, \cdots, (pq-1)\}$이라는것을 고찰하자. n과 서로 소가 아닌 나머지들은 모임$\{p, 2p, \cdots, (q-1)p\}$, 모임$\{q, 2q, \cdots, (p-1)q\}$와 0이다. 따라서

$$\begin{aligned}\phi(n) &= pq - [(q-1) + (p-1) + 1] \\ &= pq - (p+q) + 1 \\ &= (p-1) \times (q-1) \\ &= \phi(p) \times \phi(q)\end{aligned}$$

$\phi(21) = 12 = \phi(3) \times \phi(4) = 2 \times 6 = (3-1) \times (7-1)$
로서 12개의 옹근수들은 $\{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$이다.

오일레르정리

오일레르정리는 서로 소인 매 a와 n에 대하여 다음과 같이 정식화된다.

$$a^{\phi(n)} \equiv 1 \bmod n \qquad (7\text{-}5)$$

$a = 3;\ n = 10;\ \phi(10) = 4;\ 3^4 = 81 \equiv 1 \bmod 10$
$a = 2;\ n = 11;\ \phi(11) = 10;\ 2^{10} = 1024 \equiv 1$ md 11

증명: n이 씨수이면 $\phi(n)=(n-1)$ 인 경우이므로 식 7-5는 성립하며 따라서 페르마정리가 성립한다. 그러나 이것은 또한 임의의 옹근수 n에 대해서도 성립한다. $\phi(n)$은 n과 서로 소이며 n보다 작은 정의옹근수의 수이라는것을 상기하자. 다음과 같이 표시한 이런 옹근수들의 모임을 고찰하자.

$$R = \{x_1,\ x_2,\ \ldots,\ x_{\phi(n)}\}$$

이제 mod n에 관하여 매 원소들을 a로 곱하자. 즉

$$S = \{(ax_1 \bmod n), (ax_2 \bmod n), \ldots, (ax_{\phi(n)} \bmod n)\}$$

이 모임은 다음과 같은 리유로 하여 R의 치환이다.

1. a가 n과 서로 소이고 x_i가 n과 서로 소이므로 ax_i 역시 n과 서로 소이여야 한다. 이리하여 S의 원소모두는 n과 서로 소이고 n보다 작은 옹근수이다.
2. S에는 중복이 없다. 식 7-2에 귀착된다. $ax_i \bmod n = ax_j \bmod n$이면 $x_i = x_j$이다.

그러므로

$$\prod_{i=1}^{\phi(n)} (ax_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$\prod_{i=1}^{\phi(n)} ax_i \equiv \prod_{i=1}^{\phi(n)} x_i \pmod n$$

$$a^{\phi(n)} + \left[\prod_{i=1}^{\phi(n)} x_i\right] \equiv \prod_{i=1}^{\phi(n)} x_i \pmod n$$

$$a^{\phi(n)} \equiv 1 \pmod n$$

이 정리에 대한 다음과 같은 다른 한가지 형식이 또한 유용하다.

$$a^{\phi(n)+1} \equiv a \pmod n \qquad (7\text{-}6)$$

우의 따름을 RSA알고리듬의 타당성을 보여 주는데 유용한 오일레르정리에로 발전

시킬수 있다. 주어 진 두 씨수 p와 q 그리고 $0 < m < n$인 옹근수 $n=pq$와 m에 대하여 다음의 식이 성립한다.

$$m^{\phi(n)+1} = m^{(p-1)(q-1)+1} \equiv m \bmod n \qquad (7\text{-}7)$$

$\gcd(m,n)=1$ 즉 m과 n이 서로 소이면 오일레르정리(식 7-5)에 의하여 식이 성립한다. $\gcd(m,n) \neq 1$이라고 가정하자. 이것은 무엇을 의미하는가? $n=pq$이므로 식 $\gcd(m,n)=1$은 론리식(m은 p의 배수가 아니다.) AND(m은 q의 배수가 아니다.)와 동등하다. m이 p의 배수라면 n과 m은 씨인수 p를 공유하므로 서로 소가 아니며 m이 q의 배수라면 m과 n은 씨인수 q를 공유하므로 서로 소가 아니다. 그러므로 식 $\gcd(m,n) \neq 1$은 우의 론리식의 부정과 동등해야 한다. 따라서 식 $\gcd(m,n) \neq 1$은 론리식(m은 p의 배수이다.) OR (m은 q의 배수이다.)와 동등이다.

m이 p의 배수인 경우를 고찰하자. 이때는 어떤 정의옹근수 c에 대하여 식 $m=cp$가 성립한다. 이 경우에는 $\gcd(m,q)=1$이여야 한다. 그렇지 않으면 m이 p의 배수이고 또한 m이 q의 배수이므로 $m < pq$이다. $\gcd(m,q)=1$이면 오일레르정리가 성립하므로 다음과 같다.

$$m^{\phi(q)} \equiv 1 \bmod q$$

그러나 이때 Mod 산수의 규칙으로부터

$$[m^{\phi(q)}]^{\phi(p)} \equiv 1 \bmod q$$
$$m^{\phi(n)} \equiv 1 \bmod q$$

이리하여 어떤 옹근수 k가 있어서

$$m^{\phi(n)} = 1 + kq$$

이다. 량변을 $m = cp$로 곱하면

$$m^{\phi(n)+1} = m + kcpq = m + kcn$$
$$m^{\phi(n)+1} \equiv m \bmod n$$

마찬가지로 m이 q의 배수인 경우도 증명할수 있다. 이리하여 식 7-7을 증명하였다. 이 따름에 대한 다음과 같은 다른 하나의 형식이 유용하다.

$$\left[m^{\phi(n)}\right]^k \equiv 1 \bmod n$$
$$m^{k\phi(n)} \equiv 1 \bmod n \qquad (7\text{-}8)$$
$$m^{k\phi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \bmod n$$

7.4 씨수판정

큰 수가 씨수인지 아닌지를 결정하는 단순하고도 효과적인 수단은 없다. 여기서는 한가지 매력적인 방식을 제기한다. 우선 몇가지 결과들을 유도하는것이 필요하다. 첫번째는 다음과 같다.

p 가 홀씨수이면 방정식

$$x^2 \equiv 1 \pmod p$$

는 오직 2개의 풀이 즉 $x \equiv 1$ 과 $x \equiv -1$ 만을 가진다.

증명:

$$x^2 - 1 \equiv 0 \pmod p$$
$$(x+1)(x-1) \equiv 0 \pmod p$$

Mod산수의 법칙에 의하여 마지막식은 p가 $(x+1)$ 또는 $(x-1)$ 혹은 둘 다를 나눌것을 요구한다. p가 $(x+1)$과 $(x-1)$을 둘 다 나눈다고 하자. 이때 어떤 옹근수 k와 j에 대하여 $(x+1)=kp$이고 $(x-1)=jp$이라는것을 알수 있다. 두 식을 덜면 $2=(k-j)p$이다. 이 방정식은 $p=2$에 대하여서만 참으로 될수 있다. 정리에 의하여 홀씨수에 대해서만 고찰한다. 그러므로 주어 진 풀이 x에 대하여 $p|(x+1)$이거나 $p|(x-1)$이며 둘 다 성립하지는 않는다. $p|(x-1)$이라고 가정한다. 이때 어떤 k에 대하여

$$x - 1 = kp$$

이므로 $x \equiv 1 \bmod p$이다. 마찬가지로 다른 풀이에 대하여 $x \equiv -1 \bmod p$를 얻는다.

$x^2 \equiv 1 \pmod 7$	$x^2 \equiv 1 \pmod 8$
표 7-3의 ㄴ을 리용하면	표 7-2의 ㄴ을 리용하면
$1^2 \equiv 1 \bmod 7$	$1^2 \equiv 1 \bmod 8$
$6^2 \equiv 36 \bmod 7 \equiv 1 \bmod 7$; $6 \equiv -1 \bmod 7$	$3^2 \equiv 9 \bmod 8 \equiv 1 \bmod 8$
풀이: 1, -1	$5^2 \equiv 25 \bmod 8 \equiv 1 \bmod 8$; $5 \equiv -3 \bmod 8$
	$7^2 \equiv 49 \bmod 8 \equiv 1 \bmod 8$; $7 \equiv -1 \bmod 8$
	풀이: 1, -1, 3, -3

정리를 다음과 같이 다른 방식으로 정식화할수 있다. ± 1 아닌 $x^2 \equiv 1 \pmod n$의 풀이가 존재하면 n은 씨수가 아니다.

이제는 어떤 수가 씨수인지 아닌지를 검사하는 알고리듬을 정식화할수 있다 [MILL75, RABI80]. 핵심알고리듬 WITNESS는 다음과 같이 정의된다.

```
    WITNESS(a,n)
1.      b_k b_{k-1}...b_0 이 (n − 1)의 2진표현이라고 하자.
2.      d ← 1
3.      for i ← k downto 0
4.          do x ← d
```

```
5.          d ← (d × d) mod n
6.          if d = 1 이고 x ≠ 1 이며 x ≠ n − 1
7.            then return TRUE
8.          if bi = 1
9.            then d ← (d × a) mod n
10.     if d ≠ 1
11.     then return TRUE
12.     return FALSE
```

WITNESS에로의 입력은 씨수인지 아닌지를 판정해야 할 수 n과 n보다 작은 어떤 옹근수 a이다. 목적은 n이 씨수인지 아닌지를 판정하는것이다. WITNESS가 TRUE를 귀환하면 n은 확정적으로 씨수가 아니며 FALSE를 귀환하면 씨수일수도 있다.

WITNESS와 Mod산수에서의 제곱을 계산하는 그림 6-7의 알고리듬을 비교해 보면 3행부터 9행은 값 a^{n-1} mod n으로서 d를 계산한다는것을 알수 있다. n이 씨수이면 $a^{n-1} \equiv 1$ mod n이라는것이 페르마정리(식 7-3)로부터 나온다. 이리하여 d의 최종결과가 1이 아니라면 n은 씨수가 아니라는것이 나오며 TRUE를 귀환한다. 이제 6행의 판정을 고찰하자. $(n-1) \equiv -1$ mod n이므로 이 행은 ±1아닌 뿌리로 $x^2 \equiv 1$인지 아닌지를 판정한다. 앞에서 정식화한 정리에 의하여 이 조건은 n이 오직 씨수일 때만 성립한다. 이리하여 이 판정은 통과되고 WITNESS는 TRUE를 귀환한다.

그림 6-8을 다시 고찰하자. 이 경우에 n=561이고 a=7이다.
$a^{280} \equiv 67$ mod 561이고 $a^{560} \equiv 1$ mod 561이므로 WITNESS는 마지막두제곱단계에서 ±1아닌 2차뿌리를 하나 구한다. 이 시점에서 WITNESS는 TRUE를 귀환한다.

그래서 WITNESS가 TRUE를 귀환하면 수 n은 씨수가 아니다. 씨수가 아닌 홀수 n과 우연적으로 선택된 옹근수 $a < n$이 주어 졌을 때 WITNESS가 FALSE(즉 n이 씨수가 아니라는것을 판정하는것이 실패)를 귀환할 확률은 0.5이하이라는것을 보여 줄수 있다[CORM90].

이것은 일정한 신뢰성으로 홀수 n이 씨수인지 아닌지를 판정할 토대를 준다. 그 절차는 다음과 같다. 우연적으로 선택된 값 a를 리용하여 WITNESS(a,n)를 반복한다. 임의의 시점에서 WITNESS가 TRUE를 귀환하면 n은 씨수가 아니라는것이 판정된다. WITNESS가 련속적으로 s번 FALSE를 귀환하면 n이 씨수일 확률은 적어도 $1-2^{-S}$이다(CORM90,843페지를 보시오). 이리하여 충분히 큰 값 s에 대하여 n이 씨수일것이라는것을 확신할수 있다.

7.5 유클리드알고리듬

수론에서 기초기법중의 하나는 두 정의옹근수의 최대공약수를 구하는 간단한 절차인 유클리드알고리듬이다. 확장된 형태의 유클리드알고리듬은 두 정의옹근수의 최대공약수를 구하고 이 수들이 서로 소이면 다른것에 관하여 한 수의 곱하기역수도 구한다.

최대공약수구하기

유클리드알고리듬은 다음의 정리에 기초하고 있다. 임의의 부아닌 옹근수 a와 임의의 정의옹근수 b에 대하여

$$\gcd(a,b)=\gcd(b, a \bmod b) \qquad (7\text{-}9)$$

$$\gcd(55,22)=\gcd(22,55 \bmod 22)=\gcd(22,11)=11$$

이것을 보기 위하여 $d=\gcd(a,b)$라면 gcd의 정의에 의하여 $d|a$이고 $d|b$이다. 임의의 정의옹근수 b에 대하여 a를

$$a=kb+r \equiv r \bmod b$$
$$a \bmod b=r$$

형태로 표현할수 있다. 그러므로 어떤 옹근수 k에 대하여 $(a \bmod b)=a-kb$이다. 그러나 $d|b$이므로 그것은 또한 kb를 나눈다. 또한 $d|a$이다. 그러므로 $d|(a \bmod b)$이다. 이것은 d가 b와 $(a \bmod b)$의 공약수라는것을 보여 준다. 그러므로 d가 b와 $(a \bmod b)$의 공약수이면 $d|kb$와 $d|[kb+(a \bmod b)]$도 공약수이고 따라서 $d|a$와 동등하다. 이리하여 a와 b의 공약수들의 모임은 b와 $(a \bmod b)$의 공약수들의 모임과 같다. 이리하여 하나의 최대공약수는 다른것의 최대공약수와 같으며 따라서 정리는 증명된다.

식 7-9를 반복적으로 리용하여 최대공약수를 구할수 있다.

$$\gcd(18,12)=\gcd(12,6)=\gcd(6,0)=6$$
$$\gcd(11,10)=\gcd(10,1)=\gcd(1,0)=1$$

유클리드알고리듬은 식 7-9를 반복리용하여 최대공약수를 다음과 같이 구한다. 이 알고리듬은 $d > f > 0$이라는것을 전제로 한다. $\gcd(a,b)=\gcd(|a|,|b|)$이므로 알고리듬을 정의옹근수에로 제한하여도 일반성을 잃지 않는다.

```
EUCLID(d,f)
1. X←f; Y←d
2. if Y=0 return X=gcd(d,f)
3. R=X mod Y
4. X←Y
5. Y←R
6. goto 2
```

어떤 독자들은 이 처리가 완료된다는것을 어떻게 담보할수 있는가라고 물을수 있다. 즉 일정한 시점에서 Y가 X를 나눈다는것을 어떻게 믿을수 있는가? 그렇지 않다면 정의옹근수들의 무한렬이 있어서 매수는 엄격히 자기앞의 수보다 작아야 하는데 이것은 분명히 불가능하다.

gcd(1970,1066)을 구하자.	
1970=1×1066+904	gcd(1066,904)
1066=1×904+162	gcd(904,162)
904=5×162+94	gcd(162,94)
162=1×94+68	gcd(94,68)
94=1×68+26	gcd(68,26)
68=2×26+16	gcd(26,16)
26=1×16+10	gcd(16,10)
16=1×10+6	gcd(10,6)
10=1×6+4	gcd(6,4)
6=2×2+2	gcd(4,2)
2=2×2+0	gcd(2,0)
그러므로 gcd(1970,1066)=2	

곱하기역수구하기

gcd(d,f)=1이면 d는 mod f에 관하여 곱하기역수를 가진다. 즉 정의옹근수 $d<f$에 대하여 $dd^{-1}=1 \bmod f$인 $d^{-1} < f$가 존재한다. 유클리드알고리듬을 확장하여 gcd(d,f)를 구하는것외에도 최대공약수가 1이면 이 알고리듬은 d의 곱하기역수를 귀환한다.

```
EXTENDED EUCLID(d,f)
1.    (X1,X2,X3)←(1,0,f); (Y1,Y2,Y3)←(0,1,d)
2.    if Y3=0  return X3=gcd(d,f); 역수 없다.
3.    if Y3=1  return Y3=gcd(d,f); Y2=d^-1 mod f
4.    Q = ⌊X3/Y3⌋
5.    (T1,T2,T3)←(X1−QY1, X2−QY2, X3−QY3)
6.    (X1,X2,X3)←(Y1,Y2,Y3)
7.    (Y1,Y2,Y3)←(T1,T2,T3)
8.    goto 2
```

계산을 통하여 다음의 관계식이 성립한다.

$$fT1+dT2=T3 \qquad fX1+dX2=X3 \qquad fY1+dY2=Y3$$

이 알고리듬이 정확히 gcd(d,f)를 귀환한다는것을 보기 위하여 유클리드알고리듬의 X,Y를 각각 확장된 유클리드알고리듬의 X3, Y3과 같게 하면 두 변수의 취급은 같다는 것을 강조한다. 유클리드알고리듬의 매 반복에서 X는 Y의 선행값과 같게 놓는다. 마찬가지로 확장된 유클리드알고리듬의 매 단계에서 X3은 Y3의 선행값과 같게 놓으며 Y3은 X3의 선행값-Y3으로 나눈 X3의 상과 같게 놓는다. 이 마지막값은 단순히 Y3으로 나눈 X3의 나머지 X3 mod Y3이다.

또한 gcd(d,f)=1이면 최종단계에서 Y3=0이며 X3=1이라는것을 강조한다. 그러므로 선행단계에서 Y3=1이다. 그러나 Y3=1이면 다음과 같은것을 알수 있다.

fY1+dY2=Y3
fY1+dY2=1
dY2=1+(-Y1)×f
dY2≡1 mod f

그리고 Y2는 mod *f*에 관하여 *d*의 곱하기역수이다.

표 7-5는 알고리듬의 실행실례이다. gcd(550,1769)=1이며 550의 곱하기역수는 그자체 즉 550×550 ≡ 1 mod 1769이라는것을 보여 준다.

표 7-5. 확장된 유클리드(550,1769)

Q	X1	X2	X3	Y1	Y2	Y3
—	1	0	1769	0	1	550
3	0	1	550	1	-3	119
4	1	-3	119	-4	13	74
1	-4	13	74	5	-16	45
1	5	-16	45	-9	29	29
1	-9	29	29	14	-45	16
1	14	-45	16	-23	74	13
1	-23	74	13	37	-119	3
4	37	-119	3	-171	550	1

이 알고리듬에 대한 보다 구체적인 증명은 [KNUT97]에 있다.

7.6 중국나머지정리

수론에서 가장 중요한 요소의 하나는 중국나머지정리(CRT)이다(CRT는 Chinese Remainder Theorem의 략자이다). 본질에 있어서 CRT는 일정한 령역의 옹근수들을 둘씩 서로 소인 나머지들의 모임을 모드로 하는 나머지들로 재구성할수 있다는것이다.

Z_{10}의 10개 옹근수 {0,1,... ...,9}들을 (10의 서로 소인 씨인수)2와 5를 모드로 하는 두 나머지들로 재구성할수 있다. 10진수자 x의 알려 진 나머지는 r_2=0과 r_5=3 즉 x mod 2=0이고 x mod 5=3이라고 하면 x는 Z_{10}의 짝수이며 동시에 5로 나누었을 때 3이다. 유일한 풀이는 x=8이다.

CRT를 여러가지 방식으로 정식화할수 있다. 이 책의 관점에서 가장 유용한 정식화를 여기서 제기한다. 다른 하나의 정식화는 문제 7-13에서 설명한다.

$$M=\prod_{i=1}^{k} m_i$$

라고 하자. 여기서 m_i는 둘씩 서로 소이다. 즉 $1 \le i, j \le k$에 대하여 $\gcd(m_i, m_j)=1$이다. 다음과 같은 대응

$$A \leftrightarrow (a_1, a_2, \ldots, a_k) \qquad (7\text{-}10)$$

을 리용하여 Z_M의 임의의 옹근수를 k차벡토르로 표현할수 있다. 여기서 $A \in Z_M$, $a_i \in Z_{m_i}$이고 $1 \le i \le k$에 대하여 $a_i = A \bmod m_i$이다. CRT는 다음과 같은 두개의 주장을 의미한다.

1. 식 7-10의 넘기기는 Z_M과 직적 $Z_{m_1} \times Z_{m_2} \times \ldots \times Z_{m_k}$ 사이의 1:1대응이다. 즉 $0 \le A < M$인 매 옹근수 A에 대하여 $0 \le a_i < m_i$인 유일한 k차벡토르 $(a_1, a_2, \ldots, a_k)$가 있으며 이런 매 k차벡토르에 대하여 Z_m의 유일한 A가 있다.
2. Z_m의 원소들에 대하여 진행된 연산을 동등하게 대응하는 k차벡토르들우에서 진행할수 있으며 이때 매 성분들에 대한 연산은 독립적이다.

우의 두번째 주장을 다음과 같이 정식화할수 있다.

$$A \leftrightarrow (a_1, a_2, \ldots, a_k);\ B \leftrightarrow (b_1, b_2, \ldots, b_k)$$

라면

$$\begin{aligned}(A+B) \bmod M &\leftrightarrow ((a_1+b_1) \bmod m_1, \cdots, (a_k+b_k) \bmod m_k)\\ (A-B) \bmod M &\leftrightarrow ((a_1-b_1) \bmod m_1, \cdots, (a_k-b_k) \bmod m_k)\\ (A\times B) \bmod M &\leftrightarrow ((a_1\times b_1) \bmod m_1, \cdots, (a_k\times b_k) \bmod m_k)\end{aligned}$$

첫번째 주장을 고찰하자. A에서 $(a_1, a_2, \ldots, a_k)$에로의 변환은 분명히 유일하다. 왜냐하면 $a_i = A \bmod m_i$ 로 취하기때문이다. $(a_1, a_2, \ldots, a_k)$로부터 A의 계산은 다음과 같이 할수 있다. $1 \le I \le k$에 대하여 $M_i = M/m_i$ 라고 하자. $M_i = m_1 \times m_2 \times \cdots \times m_{i-1} \times m_{i+1} \times \cdots \times m_k$이므로 모든 $i \ne j$에 대하여 $M_i \equiv 0 (\bmod m_j)$이다. 이때 $1 \le i \le k$에 대하여

$$c_i = M_i \times (M_i^{-1} \bmod m_i) \qquad (7\text{-}11)$$

라고 하자. M_i의 정의로부터 그것은 m_i와 서로 소이므로 $\bmod m_i$에 관하여 유일한 곱하기역수를 가진다. 따라서 방정식 7-11은 타당하며 유일한 값 c_i를 결정한다. 이제는

$$A \equiv \left(\sum_{i=1}^{k} a_i c_i\right) \bmod M \qquad (7\text{-}12)$$

를 계산할수 있다.

식 7-12에 의해 결정된 값 A가 정확하다는것을 보기 위해서는$1 \le i \le k$에 대하여 $a_i = A \bmod m_i$이라는것을 보여 주어야 한다. $i \ne j, c_i \equiv 1 (\bmod m_i)$이면 $c_j \equiv M_j \equiv 0 (\bmod m_i)$이다.

산법연산과 관련되는 CRT의 두번째 주장은 Mod산수에 대한 규칙으로부터 나온다.

중국나머지정리의 유용한 기능의 한가지는 mod M에 관한 수(극히 큰 수)를 보다 작은 수들의 벡토르로 관리하는 방법을 주는것이다. 이것은 M이 150자리이상일 때 유용할수 있다.

973 mod 1813을 mod 37과 49에 관한 수쌍으로 표현하자.

$$m_1 = 37$$
$$m_2 = 49$$
$$M = 1813$$
$$M = 973$$

또한 M_1 = 49이고 M_2 = 37이다. 확장된 유클리드알고리듬을 리용하면 M_1^{-1} = 34 $m$od m_1이고 M_2^{-1} = 4 mod m_2이다(종국적으로는 매 M_i와 M_i^{-1}을 계산해야 한다는것을 강조한다). mod 37과 49에 관한 나머지를 취하면 973의 표현은 (11, 42)이다. 왜냐하면 973 mod 49 = 42이기때문이다.

이제 678과 973을 더하려고 한다고 가정하자. (11, 42)로 무엇을 하는가? 우선 (678) ↔ (678 mod 37, 678 mod 49)=(12, 41)을 계산한다. 이때 벡토르성분별로 더하면 (11+12 mod 37, 42+41 mod 49)=(23, 34)로 된다. 이것이 정확하다는것을 검증하기 위하여

$$\begin{aligned}(23, 34) \leftrightarrow\ & a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} \bmod M \\ &= [(23)(49)(34)+(34)(37)(4)] \bmod 1813 \\ &= 43350 \bmod 1813 \\ &= 1651\end{aligned}$$

을 계산하여 그것이 (973 + 678) mod 1813 = 1651과 같다는것을 검열한다. 1651(mod 1813)을 73으로 곱하려고 한다고 가정하자. (23, 24)를 73으로 곱하면 (23×73 mod 37, 34×73 mod 49) = (14, 32)로 된다.

$$\begin{aligned}(14, 32) \leftrightarrow\ & [(14)(49)(34) + (32)(37)(4)] \bmod 1813 \\ &= 865 \\ &= 1651 \times 73 \bmod 1813\end{aligned}$$

이라는것이 쉽게 검증된다.

7.7 리산로그

리산로그는 디피-헬만의 열쇠교환과 수자서명알고리듬(DSA)을 포함한 일련의 공개열쇠알고리듬에서 기본이다. 이 절에서는 리산로그에 대하여 간단히 개괄한다. 흥미 있는 독자들이 이에 대한 보다 구체적인것을 참고하기 위하여서는 [ORE76]과 [LEVE90]을 보시오.

mod n에 관한 제곱

오일레르정리(식 7-5)로부터 서로 소인 매 a와 n에 대하여

$$a^{\phi(n)} \equiv 1 \bmod n$$

이라는것을 상기하자. 여기서 오일레르함수 $\phi(n)$은 n과 서로 소이며 n보다 작은 정의옹근수들의 수이다. 이제 일반식

$$a^m \equiv 1 \bmod n \qquad (7\text{-}13)$$

을 고찰하자. a와 n이 서로 소이면 적어도 하나의 옹근수 m이 있어서 식 7-13을 만족시킨다. 식 7-13이 성립하는 최소의 정의제곱지수 m은 다음과 같이 여러가지 용어로 불리운다.

- a (mod n)의 차수
- a가 속하는 mod n에 관한 제곱지수
- a로 생성된 주기의 길이

우의 마지막관점에서 mod 19에 관한 7의 제곱을 고찰하자.

$$7^1 = 7 \qquad \bmod 19$$
$$7^2 = 49 = 2\times19+11 = 11 \qquad \bmod 19$$
$$7^3 = 343 = 18\times19+1 = 1 \qquad \bmod 19$$
$$7^4 = 2401 = 126\times19+7 = 7 \qquad \bmod 19$$
$$7^5 = 16807 = 884\times19+11 = 11 \qquad \bmod 19$$

렬이 반복되므로 계속할 필요는 없다. 이것은 $7^3=1 \pmod{19}$이다는것을 강조하여 증명할수 있으므로 $7^{3+j}=7^3\times7^j=7^j \pmod{19}$이며 따라서 제곱지수가 3(또는 3의 배수)아닌 7의 임의의 2개의 제곱들은 (mod 19에 관하여)서로 합동이다. 달리 말하여 렬은 주기적이며 주기의 길이는 $7^m=1 \pmod{19}$인 가장 작은 정의제곱지수 m이다. 표 7-6은 모든 정의 a<19에 대하여 mod 19에 관한 a의 제곱모두를 보여 준다. 다음과 같은것을 강조한다.

1. 모든 렬은 1로 끝난다. 이것은 앞절들의 론의에 모순되지 않는다.
2. 렬의 길이는 $\phi(19)=8$을 나눈다. 즉 렬의 완전한 수는 표의 매행에서 발생한다.
3. 어떤 렬들은 길이 18이다. 이런 경우에는 밑수 a가 mod 19에 관한 부아닌 옹근수들의 모임을 생성한다(제곱을 거쳐). 이런 매 옹근수를 mod 19에 관한 원시뿌리라고 한다.

표 7-6. mod 19에 관한 옹근수제곱

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

보다 일반적으로 수가 속할수 있는 mod n에 관한 가능한 최고의 제곱지수는 $\phi(n)$이다. 어떤 수가 이 차수이면 그것을 n의 원시뿌리라고 한다. 이 개념의 중요성은 a가 n의 원시뿌리이면 그의 제곱

$$a, a^2, \ldots, a^{\phi(n)}$$

은 mod n에 관하여 서로 다르며 모두 n과 서로 소이다. 특히 씨수 p에 대하여 a가 p의 원시뿌리이면

$$a, a^2, \cdots, a^{p-1}$$

은 mod p에 관하여 서로 다르다. 씨수 19에 대하여 그의 원시뿌리는 2, 3, 10, 13, 14와 15이다.

일부 옹근수들은 원시뿌리를 가지지 않는다. 사실 원시뿌리를 가지는 옹근수는 오직 $2, 4, p^{\alpha}$ 및 $2p^{\alpha}$형태뿐이다. 여기서 p는 임의의 홀씨수이다.

첨수

보통의 정의실수에 대하여 로그함수는 제곱함수의 역함수이다. 이와 류사한 함수가 Mod산수에도 존재한다.

보통의 로그함수의 성질을 간단히 개괄한다. 어떤 수의 로그는 어떤 정의밑수(1은 제외)가 그 수와 같게 하기 위하여 나타내야 하는 제곱으로 정의된다. 즉 밑수 x와 값

y에 대하여

$$y = x^{\log_x(y)}$$

로그의 성질은 다음과 같다.

$$\log_x(1) = 0 \qquad (7\text{-}14)$$
$$\log_x(x) = 1 \qquad (7\text{-}15)$$
$$\log_x(yz) = \log_x(y) + \log_x(z) \qquad (7\text{-}16)$$
$$\log_x(y^r) = r \times \log_x(y) \qquad (7\text{-}17)$$

어떤 씨수 p에 대한 원시뿌리 a를 고찰하자(인수는 물론 씨수가 아니라고 한다). 이때 1부터 $(p-1)$까지 a의 제곱은 정확히 한번 1부터 $(p-1)$까지의 매 옹근수로 된다. 또한 임의의 옹근수 b를 Mod산수의 정의에 의하여

$$b \equiv r \bmod p, \qquad \text{여기서 } 0 \le r \le (p-1)$$

의 형태로 표현할수 있다. 임의의 옹근수 b와 씨수 p의 원시뿌리 a에 대하여

$$b \equiv a^i \bmod p, \qquad \text{여기서 } 0 \le i \le (p-1)$$

인 유일한 제곱지수 i를 구할수 있다는것이 나온다. 이 제곱지수 i를 mod p에 관한 밑수 a의 첨수라고 한다. 이 값을 $\mathrm{ind}_{a,p}(b)$로 표시한다.

다음의 사실이 성립한다.

$$a^0 \bmod p = 1 \bmod p = 1\text{이므로 } \mathrm{ind}_{a,p}(1) = 0 \qquad (7\text{-}18)$$
$$a^1 \bmod p = a\text{이므로 } \mathrm{ind}_{a,p}(a) = 1 \qquad (7\text{-}19)$$

씨수아닌 모드를 리용하는 실례이다. $n=9$라고 하자.
$\phi(n)=6$이고 $a=2$는 원시뿌리이다. a의 각이한 제곱을 구하자.

$$2^0=1 \quad 2^4=7$$
$$2^1=2 \quad 2^5=5 \qquad (\bmod 9)$$
$$2^2=4 \quad 2^6=1$$
$$2^3=8$$

이리하여 mod 9에 관하여 뿌리 $a=2$에 대한 다음과 같은 첨수와 수들의 표를 얻는다.

첨수	0	1	2	3	4	5
수	1	2	4	8	7	5

주어 진 수의 첨수를 얻기 위하여 9와 서로 소인 나머지들이 초기입구이도록 표를 재배치한다.

수	1	2	4	5	7	8
첨수	0	1	2	5	4	3

이제

$$x = a^{\text{ind}_{a,p}(x)} \bmod p \qquad y = a^{\text{ind}_{a,p}(y)} \bmod p$$
$$xy = a^{\text{ind}_{a,p}(xy)} \bmod p$$

를 고찰하자. 모드곱하기규칙을 리용하면

$$a^{\text{ind}_{a,p}(xy)} \bmod p = (a^{\text{ind}_{a,p}(x)} \bmod p)(a^{\text{ind}_{a,p}(y)} \bmod p)$$
$$= (a^{\text{ind}_{a,p}(x)+\text{ind}_{a,p}(y)}) \bmod p$$

그러나 서로 소인 매 a와 n에 대하여

$$a^{\phi(n)} \equiv 1 \bmod n$$

이라는것을 의미하는 오일레르정리를 고찰하자. 임의의 정의옹근수 z를 $z=q+k\phi(n)$형태로 표현할수 있다. 그러므로 오일레르정리로부터

$$z=q \bmod \phi(n)\text{이면} \quad a^z=a^q \bmod n$$

이것을 앞의 등식에 적용하면

$$\text{ind}_{a,p}(xy)=[\text{ind}_{a,p}(x)+\text{ind}_{a,p}(y)] \bmod \phi(p)$$

이며 일반화하면

$$\text{ind}_{a,p}(y^r)=[r\times\text{ind}_{a,p}(y)] \bmod \phi(p).$$

이것은 실제로그와 첨수사이의 류사성을 보여 준다. 이 리유로 하여 후자를 때때로 리산로그라고도 한다.

표 7-6으로부터 직접 유도된 표 7-7은 mod 19에 관하여 정의될수 있는 리산로그들의 모임을 보여 준다.

표 7-7. mod 19에 관한 리산로그표

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$Ind_{2,19}(a)$	18	1	13	2	16	14	6	3	8	17	12	15	5	7	11	4	10	9

ㄱ) mod 19에 관하여 밑수 2에 대한 리산로그

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$Ind_{3,19}(a)$	18	7	1	14	4	8	6	3	2	11	12	15	17	13	5	10	16	9

ㄴ) mod 19에 관하여 밑수 3에 대한 리산로그

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$Ind_{10,19}(a)$	18	17	5	16	2	4	12	15	10	1	6	3	13	11	7	14	8	9

ㄷ) mod 19에 관하여 밑수 10에 대한 리산로그

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$Ind_{13,19}(a)$	18	11	17	4	14	10	12	15	16	7	6	3	1	5	13	8	2	9

ㄹ) mod 19에 관하여 밑수 13에 대한 리산로그

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$Ind_{14,19}(a)$	18	13	7	8	10	2	6	3	14	5	12	15	11	1	17	16	14	9

ㅁ) mod 19에 관하여 밑수 14에 대한 리산로그

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$Ind_{15,19}(a)$	18	5	11	10	8	16	12	15	4	13	6	3	7	17	1	2	12	9

ㅂ) mod 19에 관하여 밑수 15에 대한 리산로그

리산로그의 계산

방정식

$$y = g^x \bmod p$$

를 고찰하자. g, x와 p가 주어 지면 y를 계산하는것은 쉽다. 최악의 경우에는 x번 반복곱하여야 하며 보다 큰 효률을 얻기 위한 알고리듬들도 존재한다.

그러나 y, p와 g가 주어 졌을 때 일반적으로 x(리산로그)를 구하는것은 대단히 힘들다. 이 어려움성은 RSA에 요구된 씨인수분해와 같을것이다. 지금 이 시각에 모드씨수에 관하여 리산로그를 취하는 점근적인 고속알고리듬은 [BETH91]로서

$$e^{((\ln p)^{1/3} \ln(\ln p))^{2/3}}$$

인데 큰 씨수에 대해서는 적합하지 않다.

참고문헌

독자들이 바라는것보다 더 구체적인것을 서술한 수론에 관한 많은 책들이 있다. 초보적이지만 아주 유용하고 간단한 책은 [ORE97]이다. 보다 심도 있는 책을 요구하는 독자들에게는 두개의 좋은 책 [KUMA98]과 [ROSE93]이 적합하다. [LEVE90]도 물론 읽을수 있다. 모든 책들에는 자습할수 있도록 문제와 풀이를 제시하였다.

수론에 대한 기본적인 지식을 습득하기 위한 가장 좋은 책은 [BURN97]이다. 이 책은 오직 문제와 풀이들만으로 이루어 졌는데 수론에 관한 지식들을 구체적으로 전개하였다. 그러므로 모든문제를 풀면 수론에 대한 주간생과정을 마치는것과 같다.

BURN97 Burn, R. A *Pathway to Number Theory*. Cambridge, England: Cambridge University Press, 1997.

KUMA98 Kumanduri,R. ,and Romero,C. *Number Theory with Computer Applications*. Upper Saddle River,NJ:Prentice Hall,1998.

LEVE90 Leveque, W. *Elementary Theory of Numbers*. New York:Dover, 1990.

ORE67 Ore, O. *Invitation to Number Theory*. Washington, DC:The Mathematical Association of America, 1967.

ROSE93 Rosen, K. *Elementary Number Theory and its Applications*. Reading, MA: Addition-Wesley, 1993.

문 제

1. 이 문제의 목적은 두 란수가 서로 소일 확률은 0.6정도이라는 6.2에서 나오는 주장을 론증하는것이다.
   ㄱ) $P = \Pr[\gcd(a, b) = 1]$이라고 하자. $\Pr[\gcd(a, b) = d] = P/d^2$이라는것을 증명하시오.

   암시: 량 $\gcd\left(\frac{a}{b}, \frac{b}{d}\right)$를 고찰하시오.

   ㄴ)우의 문제에서 가능한 모든 d값에 대하여 결과들의 합은 1이다. 즉

$$\sum_{d \geq 1} \Pr[\gcd(a,b) = d] = 1$$

   이 등식을 리용하여 P의 값을 결정하시오. 암시: 항등식 $\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$ 을 리용하시오.
2. 두 린접한 옹근수 n과 n+1에 대하여 왜 $\gcd(n,\ n+1) = 1$인가?
3. 7.2에서는 합동관계를 다음과 같이 정의하였다. (a mod n)=(b mod n)이면 두 옹근수 a와 b는 mod n에 관하여 합동이라고 한다. 이때 $n|(a-b)$이면 $a \equiv b$ mod n이라는것을 증명하였다. 수론에 대한 일부 문헌들에서는 이 마지막식을 합동관계의 정의로 리용한다. 즉 $n|(a-b)$이면 두 옹근수 a와 b는 mod n에 관하여 합동이라고 한다. 이 마지막정의를 출발점으로 하면 (a mod n)=(b mod n)으로부터 n은 ($a-b$)를 나눈다는것을 증명하시오.

4. 다음의것들을 증명하시오.
ㄱ) $(a \bmod n)=(b \bmod n)$이면 $a \equiv b \bmod n$이다.
ㄴ) $a \equiv b \bmod n$이면 $b \equiv a \bmod n$이다.
ㄷ) $a \equiv b \bmod n$이고 $b \equiv c \bmod n$이면 $a \equiv c \bmod n$이다.
5. 다음의것들을 증명하시오.
ㄱ) $[(a \bmod n)-(b \bmod n)] \bmod n=(a-b) \bmod n$
ㄴ) $[(a \bmod n)\times(b \bmod n)] \bmod n=(a\times b) \bmod n$
6. 페르마정리를 리용하여 $3^{201} \bmod 11$을 계산하시오.
7. 고대중국수학자들은 n이 씨수이기 위해서는 n이 (2^n-2)를 나누는것이 필요하고 충분하다는 그릇된 주장을 내놓았다.
ㄱ) 홀씨수를 리용하여 이 조건을 만족시키는 실례를 드시오.
ㄴ) $n=2$일 때 이 조건은 분명히 참이다. if조건을 증명하여 n이 홀씨수이면 이 조건은 참이라는것을 증명하시오.
ㄷ) 씨수도 아니며 이 조건도 만족시키지 않는 홀수 n의 실례를 만드시오. 이것을 씨수 아닌 아주 큰 값으로 할수 있다. 이것이 바로 중국수학자들이 이 조건이 참이면 n이 씨수라고 생각한 착오였다.
ㄹ) 불행하게도 고대중국인들은 $n=341$에 대해서는 시도하지 않았다. 이 n은 씨수가 아니며$(341=11\times31)$ 또한 나머지 없이 $2^{341}-2$를 나눈다. $2^{341} \equiv 2 \bmod 341$이라는것을 오직 if조건만을 론박하여 증명하시오. 암시: 2^{341}을 계산할 필요는 없으며 그대신에 합동성을 리용하시오.
8. 표 7-4에서 $\phi(n)$은 $n>2$에 대하여 짝수이다. 이것은 모든 $n>2$에 대하여 성립한다. 이것이 왜 그런가를 간단히 설명하시오.
9. ㄱ) gcd(24140, 16762)를 구하시오.
ㄴ) gcd(4655, 12075)를 구하시오.
10. 이 문제의 목적은 유클리드알고리듬의 반복회수의 웃한계를 설정하는것이다.
ㄱ) $m=gn+r$라고 가정하자. g와 r는 부아닌 옹근수이며 $0\le r < n$이다. $m/2 > r$이라는것을 증명하시오.
ㄴ) X_i는 i번째 반복후에 유클리드알고리듬의 X값이라고 하자.

$$X_{i+2}<\frac{X_i}{2}$$

이라는것을 증명하시오.
ㄷ) m, n과 N이 $1\le m$, $\le 2^N$인 옹근수이면 유클리드알고리듬은 기껏해야 $2N$단계내에 $\gcd(m,n)$을 구한다는것을 증명하시오.
11. 유클리드알고리듬은 2000년전에 알려 졌으나 항상 수론에서 유명한것이였다. 수천년이 지난후 1961년에 제이 스테인(J. Stein)이 새로운 알고리듬을 내놓았다. 스테인의 알고리듬은 다음과 같다. $A, B>1$인 $\gcd(A,B)$를 구하자.
단계1. $A_1=A$, $B_1=B$, $C_1=C$로 놓는다.
단계n. ① $A_n=B_n$이면 정지하고 $\gcd(A,B)=A_n\times C_n$.
② A_n과 B_n이 둘 다 짝수이면 $A_{n+1}=An/2$, $B_{n+1}=B_n/2$, $C_{n+1}=2C_n$으로 놓는다.
③ A_n이 짝수이고 B_n이 홀수이면 $A_{n+1}=A_n/2$, $B_{n+1}=B_n$, $C_{n+1}=C_n$으로 놓는다.
④ A_n이 홀수이고 B_n이 짝수이면 $A_{n+1}=A_n$, $B_{n+1}=B_n/2$, $C_{n+1}=C_n$으로 놓는다.
⑤ A_n과 B_n이 둘 다 홀수이면 $A_{n+1}=|A_n-B_n|, B_{n+1}=\min(B_n,A_n), C_{n+1}=C_n$으로 놓는다.

단계 n+1을 계속한다.

ㄱ) 두 알고리듬의 차이를 보기 위하여 gcd(2152,764)를 해당한 알고리듬을 리용하여 계산하시오.

ㄴ) 유클리드알고리듬보다 스테인알고리듬의 우점은 무엇인가?

12. ㄱ) 스테인알고리듬이 n번째 단계전에 정지하지 않으면

$$C_{n+1}\times\gcd(A_{n+1},B_{n+1})=C_n\times\gcd(A_n,B_n)$$

이라는것을 증명하시오.

ㄴ) 이 알고리듬이 단계 (n−1)전에 정지하지 않는다면

$$A_{n+2}B_{n+2} \le \frac{A_nB_n}{2}$$

이라는것을 증명하시오.

ㄷ) 1≤ A, B ≤ 2^N이면 스테인알고리듬은 기껏해야 $4N$단계내에 gcd(m, n)을 구한다는것을 증명하시오. 이리하여 스테인알고리듬은 유클리드알고리듬과 같은 회수로 거칠게 동작한다.

ㄹ) 스테인알고리듬은 실제로 gcd(A,B)를 귀환한다는것을 론증하시오.

13. 중국나머지정리의 공통적인 정식화는 다음과 같다. $m_1,\dots\dots,m_k$가 $1\le i$, $j\le k$와 $i\ne j$에 대하여 둘씩 서로 소인 옹근수들이라고 하자. M을 m_i모두의 적이라고 한다. a_1, ..., a_k를 옹근수라고 하자. 이때 합동식들의 모임

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

은 mod M에 관하여 유일풀이를 가진다. 이런 형태로 정식화된 정리는 참이라는것을 증명하시오.

14. CRT를 례증한 실례는

$$x \equiv 2 \pmod 3;\ x \equiv 3 \pmod 5;\ x \equiv 2 \pmod 7$$

이였다. x를 구하시오.

15. 6명의 교수들이 각각 월, 화, 수, 목, 금, 토요일에 강의를 시작하며 각각 2, 3, 4, 1, 6, 5일동안의 강의계획이 발표되였다. 일요일에는 강의를 하지 않는다(따라서 일요일 강의는 빼야 한다).
6명의 교수모두가 언제 강의를 그만두게 되는가? 암시: CRT를 리용하시오.

16. 25의 원시뿌리모두를 구하시오.

17. 29의 원시뿌리 2가 주어 졌을 때 첨수표를 만들고 그것을 리용하여 다음의 합동식을 푸시오.

ㄱ) $17x^2 \equiv 10 \bmod 29$

ㄴ) $x^2-4x-16 \equiv 0 \bmod 29$

ㄷ) $x^7 \equiv 17 \bmod 29$

제8장. 통보문인증과 하쉬함수

아마도 망보안기술에서 가장 복잡한 분야는 통보문인증과 수자서명이라고 해야 할것이다. 공격과 대응책이 서로 몹시 얽히여 이 분야의 실천가들에게 모든 우연적인 사건들을 설명하기 위한 시도로서 돌고 있는 원우에 돌고 있는 더 큰 원을 덧놓아 보려고 한 고대의 한 천문학자를 상기시키기 시작하였다. 다행히도 이 옛날의 망각된 천문학자와는 달리 오늘의 암호규약설계자들은 근본적으로 정확한 모형으로부터 출발하여 작업하고 있는것 같다. 한권의 책에 통보문인증과 수자서명을 제기하거나 실현하는 암호학적기능과 규약들을 다 렬거하는것은 불가능하다. 그 대신에 이 장과 다음 두개 장의 목적은 이 문제에 대한 폭 넓은 개괄을 주고 여러가지 입문들을 서술하는 체계적인 방법들을 개발하는것이다.

이 장은 렬거된 인증 및 수자서명에 대한 요구의 초보와 공격의 류형부터 시작한다. 그다음에 점차적으로 안전한 하쉬함수의 중요한 분야들을 포괄하는 기본방식들을 개괄한다. 특수한 하쉬함수들은 9장에서 설명한다.

8.1 인증요구

망을 통한 통신에서는 다음과 같은 공격들을 식별할수 있다.

1. **로출(disclosure):** 적당한 암호열쇠를 소유하지 못한 사람에게나 처리에서 통보문내용의 루실.
2. **통신량(traffic)분석:** 부분들사이의 통신량패턴의 발견. 련결지향응용에서는 련결의 빈도수와 기간을 결정해야 한다. 련결지향 또는 련결 없는 환경에서 부분들사이의 통보문의 수와 길이를 결정해야 한다.
3. **가장(masquerade):** 부정적인 원천지로부터 망안에 통보문의 삽입. 이것은 인증된 실체로부터 오는것을 의미하는 적에 의한 통보문의 창조를 포함한다. 또한 통보문수신자가 아닌 다른 어떤 사람에 의하여 통보문수신 또는 비수신의 인정이 부정적이라는것을 포함한다.
4. **내용부변경:** 삽입, 삭제, 전위 및 변경을 포함하는 통보문내용의 변화.
5. **렬변경:** 삽입, 삭제, 전위 및 재정돈을 포함하는 부분들사이의 통보문렬에 대한 임의의 변경.
6. **시간관계변경:** 통보문의 지연 또는 재현. 련결지향응용에서는 완전한 대화 또는 통보문렬들은 어떤 선행한 타당한 대화의 재현이여야 하거나 렬의 개별적통보문들은 지연되거나 재시동되여야 한다. 련결 없는 응용에서는 개별적자료들이 지연되거나 재현되여야 한다.
7. **거절(repudiation):** 목적지에 의한 통보문수신의 부인 또는 원천지에 의한 통보문전송의 부인.

첫 두개 공격을 취하는 척도는 통보문기밀성의 령역에 있으며 1편에서 취급되였다.

항목 3부터 6까지의 사항들을 취급하는 척도는 일반적으로 통보문인증으로 고찰된다. 특별히 항목 7을 취급하는 기구는 수자서명의 머리부에 놓인다. 일반적으로 수자서명기술은 역시 항목 3부터 6까지의 렬거된 일부 또는 모든 항목들을 고려할것이다.

개괄적으로 말하여 통보문인증은 수신된 통보문이 주장된 원천지로부터 오며 변경되지 않았다는것을 검증하는 절차이다. 통보문인증은 또한 중단(sequencing)과 시기적절함(timeliness)을 검증할수도 있다. 수자서명은 원천지 또는 목적지에 의한 거절에 대항하는 척도들도 포함하는 인증기술이다.

8.2 인증함수

임의의 통보문인증이나 수자서명을 기본적으로 두가지 수준으로 개괄할수 있다. 낮은 수준에서는 인증자 즉 통보문을 인증하는데 리용되는 값을 생성해 내는 어떤 종류의 함수가 있어야 한다. 다음으로 이 낮은 수준의 함수는 수신자가 통보문의 인증성을 검증할수 있는 고수준인증규약에서 초기값으로 리용된다.

이 절에서는 인증자를 생성해 내는데 리용되는 함수들의 형태를 고찰한다. 이것들은 다음과 같이 3개 부류로 나눌수 있다.

- **통보문암호화:** 완전한 통보문의 암호문이 인증자로서 종사한다.
- **통보문인증부호(MAC):** 인증자로서 종사하는 고정길이값을 내보내는 통보문과 비밀열쇠에 관한 공개함수
- **하쉬함수:** 임의의 길이의 통보문을 인증자로서 종사하는 고정길이하쉬값으로 넘기는 공개함수

이제 매개에 대하여 간단히 설명한다. MAC와 하쉬함수들은 8.3와 8.4에서 더 구체적으로 설명한다.

통보문암호화

통보문암호화자체는 인증의 방법을 제공한다. 이 분석은 전통암호방식 및 공개열쇠암호방식과는 차이난다.

전통암호

전통암호의 간단한 쓰임을 고찰하자(그림 8-1의 ㄱ). 원천지 A에서 목적지 B에로 전송된 통보문을 A와 B가 공유한 비밀열쇠 K를 리용하여 암호화한다. 이 열쇠를 아는 다른 사람이 없다면 기밀성이 담보된다. 즉 통보문의 평문을 발견할수 없다.

그외에도 B는 통보문이 A에 의해 생성되였음을 확신한다고 볼수 있다. 왜 그런가? A는 K를 소유한 유일한 다른 사람이며 따라서 K에 의해 복호화될수 있는 암호문을 구성하는데 필요한 정보를 가진 유일한 다른 사람이기때문에 통보문은 A로부터 와야 한다. 더 나아가서 M이 발견된다면 K를 모르는 적은 암호문에서 요구하는 변화를 생성해 내는 암호문의 비트들을 어떻게 변경시키였는가를 모르므로 B는 M의 그 어떤 비트도 변경되지 않았다는것을 안다.

그래서 전통암호는 기밀성은 물론 인증을 제공한다. 그러나 보통의 명제를 제한시키는것이 요구된다.

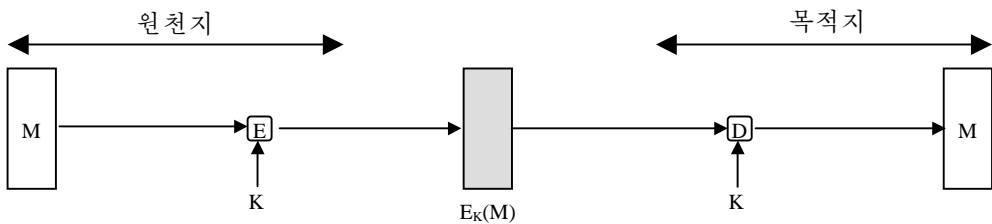
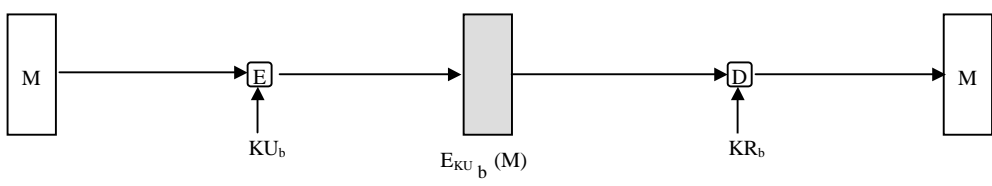
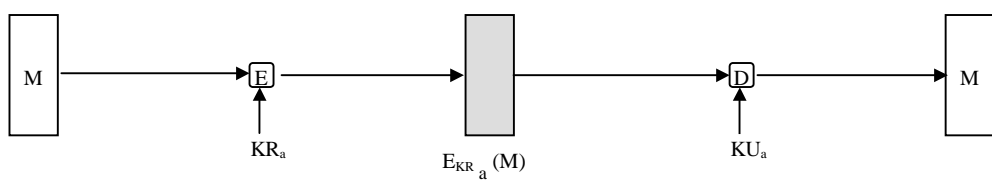
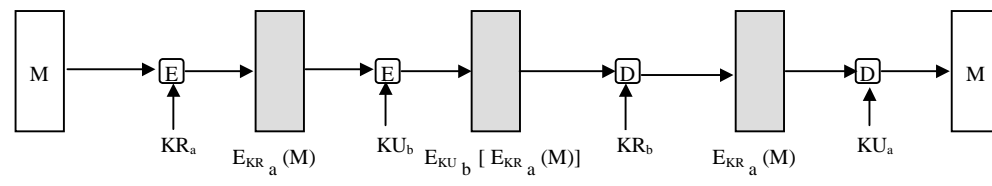
정확히 B에서 무슨 현상이 생기는가를 고찰하자. 복호화함수 D와 비밀열쇠 K가 주어 졌을 때 목적지는 임의의 입력 X를 접수하여 출력 $Y=D_K(X)$를 생성해 낼것이다. X가 대응하는 암호문에 의해 생성된 정당한 통보문 M의 암호문이라면 Y는 어떤 평문통보문 M이다. 그렇지 않으면 Y는 의미 없는 비트렬일것이다. Y가 정당한 평문이며 따라서 A로부터 와야 하는가 아닌가를 B에서 결정하는 어떤 자동화된 수단이 있을것을 요구한다.

ㄱ) 전통암호: 기밀성과 인증

ㄴ) 공개열쇠암호: 기밀성

ㄷ) 공개열쇠암호: 인증과 서명

ㄹ) 공개열쇠암호: 기밀성과 인증,서명

그림 8-1. 통보문암호화의 기본리용

인증의 관점에서 볼 때 앞절들에서의 추리를 따르는것이 자연스럽다. 통보문 M이 임의의 비트패턴일수 있다고 가정하자. 이 경우에 들어 오는 통보문이 정당한 통보문의 암호문인지 아닌지를 목적지에서 자동적으로 결정하는 방법이 없다. 이 결론은 론의할 여지가 없다. 즉 M이 임의의 비트패턴이면 X의 값을 고려함이 없이 $Y=D_K(X)$는 어떤 비트패턴이며 따라서 인증되는 평문으로 접수되여야 한다.

이리하여 일반적으로 가능한 비트패턴모두의 어떤 유일하게 작은 부분모임만이 정당한 평문으로 고찰된다. 이 경우는 임의의 가짜 암호문이 정당한 평문을 생성해 내는데 적합하지 않다. 실례로 오직 하나의 비트패턴만이 10^6에 의해 정당한 평문이라고 가정하자. 이때 임의의 우연적으로 선택된 암호문으로서 취급되는 비트패턴이 정당한 평문통보문을 생성해 낼 확률은 단지 10^{-6}이다.

일련의 응용과 암호방식에 대하여 요구된 조건은 더 말할것없이 효과적이다. 가령 하나의 밀기($K=1$)를 가진 씨저(Caesar)암호를 리용하여 영어통보문을 전송하고 있다고 가정하자. A는 다음과 같은 정당한 암호문을 보낸다.

nbsftfbupbutboeepftfbupbutboemjuumfmbnctfbujwz

B는 다음과 같은 평문을 생성해 내기 위하여 복호화한다.

marseatoatsanddoeseatoatoatsandlittlelambseativy

단순한 빈도수해석은 이 통보문이 보통영어의 륜곽을 가진다는것을 확인한다. 다른 한편 적이 다음과 같은 우연문자렬

zuvrsoevgqxlzwigamdvnmhpmccxiuureosfbcebtqxsxq

을 생성한다면 이것은 보통영어의 륜곽과는 어울리지 않는

ytuqrndufpwkyvhfzlcumlgolbbwhttqdnreabdaspwrwp

로 복호화한다.

들어 온 암호문이 의미 있는 평문으로 복호화되면 자동적으로 결정하는것은 힘들수 있다. 평문이 2진형의 파일이거나 수자화된 X-선이라면 적당히 형태화되고 인증된 평문을 결정하는것은 힘들수 있다. 이리하여 적은 정당한 사용자로부터 오는것을 의미하는 우연적인 내용과 통보문을 발행함으로써 일정한 수준의 분렬(disruption)을 달성할 수 있다.

이 문제에 대한 한가지 풀이는 쉽게 식별되지만 암호화함수에 대한 재학습이 없이는 부본을 뜰수 없는 어떤 구조를 가진 평문에 집중하는것이다. 실례로 그림 8-2의 ㄱ에서 보여 주는것처럼 암호화하기전에 매 통보문에 틀검사렬(FCS) 또는 검사합이라고 알려진 오유검출부호를 첨가할수 있다. A는 통보문 M을 준비한 다음에 이것을 FCS를 생성해 내는 함수 F에 입력으로 준다. FCS를 M에 첨가한 다음에 완전한 블로크를 암호화한다. 목적지에서 B는 FCS를 재생성하기 위해 시도하는 같은 함수 F를 적용한다. 계산된 FCS가 들어 온 FCS와 같으면 통보문을 인증된것으로 고찰한다. 임의의 우연비트렬이 요구하는 관계를 만족시키는것은 드물다.

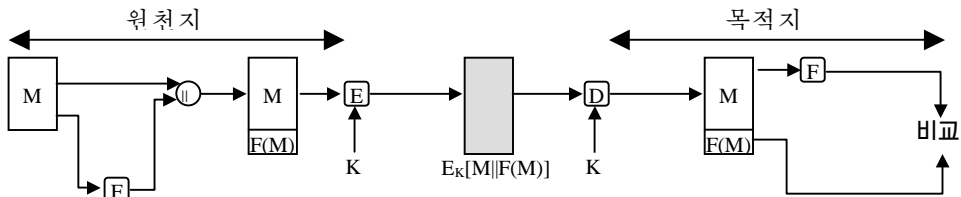
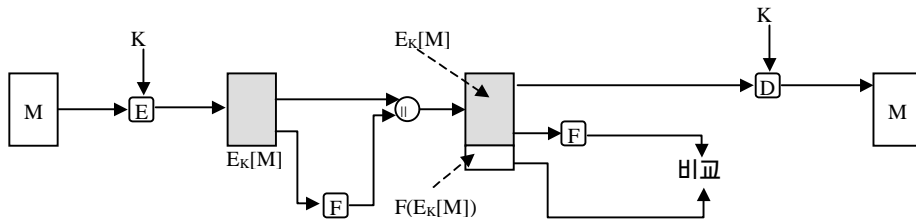



그림 8-2. 내부 및 외부조종

FCS와 암호화함수를 수행하는 순서가 결정되였다는것을 강조한다. 그림 8-2의 ㄱ에서 보여 주는 렬은 내부오유조종으로서 [DIFF79]에서 참조할수 있는데 여기서 저자는 외부오유조종(그림 8-2의 ㄴ)에 배치되게 하였다. 내부오유조종에서는 인증을 준다. 왜냐하면 적은 복호화할 때 오유조종비트들을 타당하게 하는 암호문을 생성하는것이 힘들기때문이다. 그 대신에 FCS가 바깥부호이면 적은 타당한 오유조종부호를 가진 통보문을 구성할수 있다. 적은 비록 복호화된 평문이 무엇인지 모를수 있다고 하여도 그는 아직

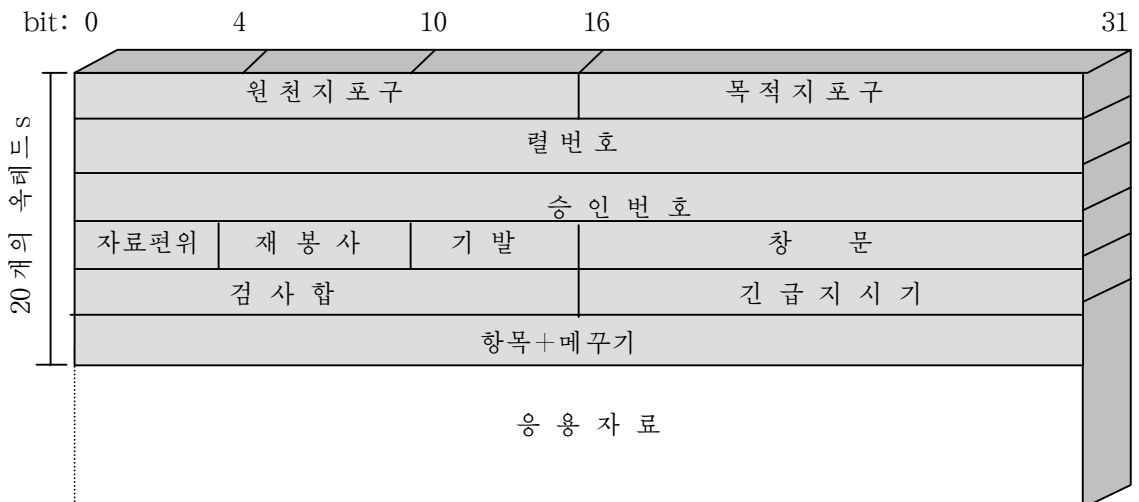



그림 8-3. TCP토막

혼동 및 분렬조작을 창조하려고 할수 있다.

오유조종부호가 바로 하나의 실례이다. 사실 전송된 통보문에 첨가된 임의의 종류의 구조는 인증능력을 강하게 하는데 종사한다. 이런 구조를 계층규약을 구성하는 통신방식의 사용자가 준다. 실례로 TCP/IP규약방식을 리용하여 전송된 통보문의 구조를 고찰하자. 그림 8-3은 TCP머리부를 보여 주는 TCP토막의 양식이다.

이제 매 주콤퓨터쌍은 유일한 비밀열쇠쌍을 공유하며 따라서 적용을 고려함이 없이 주콤퓨터쌍사이의 모든 교환은 같은 열쇠를 리용하였다고 가정하자. 이때 IP머리부를 제외한 자료묶음모두를 단순하게 암호화한다(그림 5-5를 보시오). 다시 적이 암호화된 TCP토막을 위하여 어떤 임의의 비트패턴을 치환하였다면 결과평문은 의미 있는 머리부를 포함하지 않을것이다. 이 경우에 머리부는(머리부를 덮는) 검사합뿐아니라 렬번호와 같은 다른 유용한 정보들도 포함한다. 주어 진 련결우에서 련속적인 TCP토막들이 차례로 렬거되므로 암호화는 적이 임의의 토막을 지연, 순서빠짐 또는 삭제를 하지 못한다는것을 담보한다.

공개열쇠암호

공개열쇠암호의 간단한 리용(그림 8-1의 ㄴ)은 기밀성은 담보하나 인증은 그렇지 못하다. 원천지(A)는 목적지(B)의 공개열쇠 KU_b를 리용하여 M을 암호화한다. 오직 B만이 대응하는 비밀열쇠 KR_b를 가지고 있으므로 B만이 통보문을 복호화할수 있다. 이 방식은 인증을 제공하지 못한다. 왜냐하면 임의의 적은 역시 B의 공개열쇠를 리용하여 A이라는것을 주장하는 통보문을 암호화하기때문이다.

인증을 담보하기 위하여 A는 자기의 비밀열쇠를 리용하여 통보문을 암호화하고 B는 A의 공개열쇠를 리용하여 복호화한다(그림 8-1의 ㄷ). 이것은 전통암호의 경우처럼 같은 류형의 추리를 리용하여 인증을 담보한다. 그러므로 통보문 A로부터 와야 한다. 왜냐하면 A는 KR_a를 소유한 유일한 부분이므로 KU_a로 복호화될수 있는 암호문을 구성하는데 필요한 정보를 가진 유일한 부분이다. 다시 앞에서와 같은 추리를 적용한다. 따라서 평문에 대한 어떤 내부구조가 있어서 수신자는 타당한 평문과 우연비트를 식별할수 있어야 한다.

이런 구조가 있다고 가정하면 그림 8-1의 ㄷ의 방식은 인증을 담보한다. 이것은 또한 수자서명으로서 알려 진것이 무엇인가를 보여 준다(이것은 보게 되는바와 같이 수자서명을 구성하는 방식은 아니지만 원리는 같다).

오직 A만이 암호문을 구성할수 있다. 왜냐하면 A만이 KR_a를 소유하기때문이다. B가 아니라고 하여도 수신자는 암호문을 구성하였을것이다. 그러므로 B가 암호문을 소유하면 B는 통보문이 A로부터 온다는것을 증명하는 수단을 가진다. 사실 A는 자기의 비밀열쇠를 리용하여 암호화함으로써 《서명》하였다.

이 방식은 기밀성을 제공하지 못한다는것을 강조한다. A의 공개열쇠를 가진 임의의 사람은 암호문을 복호화할수 있다.

기밀성과 인증을 둘 다 제공하기 위하여 A는 우선 수자서명을 제공하는 자기의 비밀열쇠를 리용하여 M을 암호화하고 그다음에는 기밀성을 제공하는 B의 공개열쇠를 리용하여 M을 암호화할수 있다(그림 8-1의 ㄷ). 이 방식의 불합리점은 복잡한 공개열쇠알고리듬을 매 통신마다 2번이 아니라 4번 진행해야 한다는것이다.

표 8-1은 통보문암호화에 대한 이런 여러가지 방식들의 기밀성과 인증관계를 개괄한다.

통보문인증부호

또 다른 하나의 인증기술은 통보문에 보충된 암호학적검사합 또는 MAC라고 알려진 작은 고정된 크기의 자료블로크를 생성하는데 비밀열쇠를 리용하는것이다. 이 기술은 두 통신부분 말하자면 A와 B가 공통의 비밀열쇠 K를 공유한다고 가정한다. A가 B에게 통보문을 보낼 때 그는 통보문과 열쇠에 관한 함수로서 MAC 즉 $MAC=C_K(M)$을 계산한다. 통보문+MAC는 지향된 수신자에게 전송된다. 수신자는 같은 비밀열쇠를 리용하여 새로운 MAC를 생성함으로써 수신된 통보문에 대하여 같은 계산을 진행한다. 수신된 MAC는 계산된 MAC와 비교된다(그림 8-4의 ㄱ). 오직 송수신자들만이 비밀열쇠의 신원을 안다고 가정하고 수신된 MAC가 계산된 MAC와 정합된다면

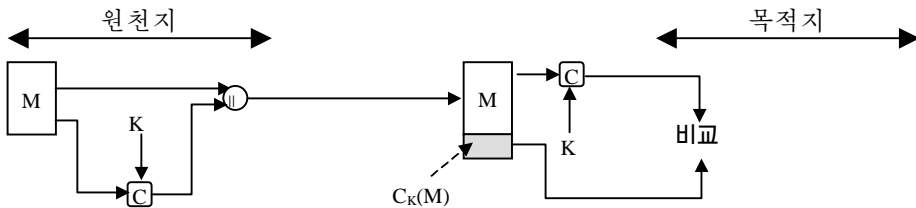
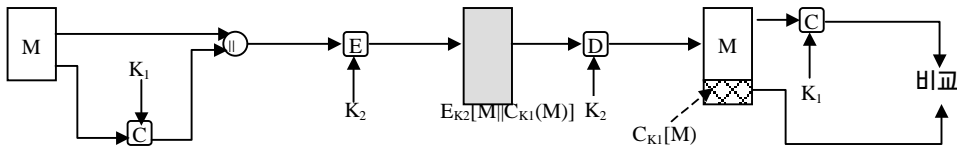
표 8-1. 통보문암호화의 기밀성과 인증관계

ㄱ) 전통(대칭)암호
A→B: $E_K[M]$ • 기밀성을 제공한다. -오직 A와 B만이 K를 공유한다. • 인증의 정도를 제공한다. -A로부터만 올수 있다. -림시적으로 변경되지 않는다. -일정한 양식/파잉성을 요구한다. • 서명을 제공하지 못한다. -수신자는 통보문을 위조할수 있다. -송신자는 통보문을 부정할수 있다.
ㄴ) 공개열쇠(비대칭)암호
A→B: $E_{KU_b}[M]$ • 기밀성을 제공한다. -오직 B만이 복호화하는 KU_b를 가진다. • 인증을 제공하지 못한다. -임의의 부분은 KU_b를 리용하여 통보문을 암호화하고 A임을 주장할수 있다.
A→B: $E_{KR_a}[M]$ • 인증과 서명을 제공한다. -오직 A만이 암호화하는 KR_a를 가진다. -림시적으로 변경하지 못한다. -일정한 양식/파잉성을 요구한다. -임의의 부분이 KU_a를 리용하여 서명을 검증할수 있다.
A→B: $E_{KU_b}[E_{KR_a}[M]]$ • KU_b때문에 기밀성을 제공한다. • KR_a때문에 인증과 서명을 제공한다.

1. 수신자는 통보문이 변경되지 않았음을 담보 받는다. 공격자가 통보문을 변경하였으나 MAC를 변경하지 못하였다면 MAC에 대한 수신자의 계산은 수신된

MAC와 차이날것이다. 공격자는 비밀열쇠를 모른다고 가정하므로 통보문에서 변경에 대응하는 MAC를 변경할수 없다.

2. 수신자는 통보문이 주장된 송신자로부터 온다는것을 담보 받는다. 송수신자들을 제외한 누구도 비밀열쇠를 모르므로 송수신자들을 제외한 누구도 고유한 MAC를 가진 통보문을 준비할수 없다.
3. 통보문이 렬번호(HDLC, X. 25와 TCP로 리용된것과 같은것)를 포함하면 공격자가 련속적으로 렬번호를 변경시킬수 없으므로 수신자는 고유한 렬임을 담보받을수 있다.

MAC함수는 암호화와 류사하다. 한가지 차이는 MAC알고리듬은 가역불가능일것을 요구한다. 왜냐하면 복호화때문이다. 인증함수의 수학적성질로부터 이것은 암호화보다 더 적게 파괴될수 있다.

ㄱ) 통보문인증

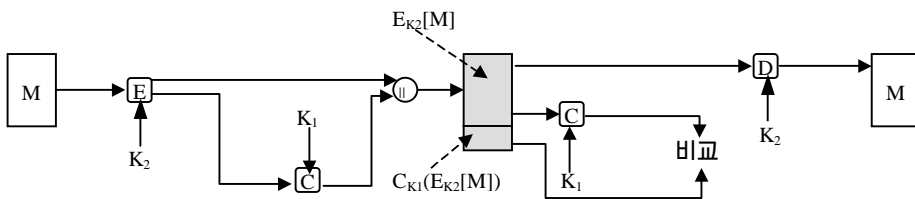
ㄴ) 통보문인증과 기밀성: 평문에 대한 인증

ㄷ) 통보문인증과 기밀성: 암호문에 대한 인증

그림 8-4. 통보문인증부호(MAC)의 기본리용

바로 묘사된 처리는 인증은 제공하나 기밀성은 제공하지못한다. 왜냐하면 전체로서의 통보문은 명백히 전송되기때문이다. MAC알고리듬을 실시한 다음에(그림 8-4의 ㄴ) 또는 전에(그림 8-4의 ㄷ) 통보문암호화를 진행함으로써 기밀성을 담보할수 있다. 이 두가지 경우 다 2개의 개별적열쇠들이 요구되며 매 열쇠들은 송수신자들에 의하여 공유된

다. 첫번째 경우에 MAC는 통보문을 입력으로서 계산하며 그다음에는 통보문에 련접시킨다. 이때 완전한 블로크가 암호화된다. 두번째 경우에는 우선 통보문을 암호화한다. 그다음에 결과암호문을 리용하여 MAC를 계산하며 전송된 블로크를 형성하는 암호문에 련접시킨다. 전형적으로 인증을 직접 평문에 련관시키여 그림 8-4의 ㄷ의 방법을 리용하는것이 적합할수 있다.

전통암호는 인증을 제공하며 쉽게 만들어 널리 리용할수 있는데 개별적통보문인증부호대신에 이것을 왜 단순하게 리용하지 못하는가? [DAVI89]는 통보문인증부호를 리용하는 다음과 같은 3개 방안을 제기하였다.

1. 일련의 목적지들에 같은 통보문을 광고하는 일련의 적용들이 있다. 실례로 망을 이제는 리용할수 없음을 사용자들에게 통보하는것 또는 군사조종쎈터에서의 경보신호를 통보하는것이다. 인증을 조종하기 위하여 응답가능한 오직 하나의 목적지만을 가지는것은 더 경제적이고 더 신뢰적이다. 이리하여 통보문은 대응된 통보문인증부호를 가지는 평문안에서 공개되여야 한다. 응답체계는 비밀열쇠를 가지며 인증을 진행한다. 위반되였다면 다른 목적체계들을 일반적인 경보로 변경한다.
2. 다른 하나의 가능한 씨나리오는 한쪽에 무거운 부담을 주어 들어 온 모든 통보문을 복호화하는 시간을 줄수 없게 하는 교환이다. 인증은 선택적인 토대와 검열을 위해 우연적으로 선택된 통보문들에 대하여 진행한다.
3. 평문으로서 콤퓨터프로그람의 인증은 아주 좋은 봉사이다. 매번 콤퓨터프로그람을 복호화함이 없이는 그것을 실행할수 없는데 이것은 처리기자원의 랑비가 많다. 그러나 통보문인증부호가 프로그람에 첨부되였다면 프로그람의 완정성을 담보하는 요구가 제기될 때마다 검사할수 있다.

다음과 같은 3개의 다른 관련식들을 첨부한다.

4. 어떤 적용에 대하여 통보문을 비밀로 간수하지는 않지만 통보문을 인증하는것이 중요하다. 그러한 실례는 단순망관리규약판 3(SNMPv3)인데 이것은 기밀성과 인증에 관한 함수들을 분리한다. 이 적용에 대하여 들어 온 SNMP통보문이 관리체계에서 파라메터를 변경시키는 지령을 포함한다면 관리체계가 특별히 들어 온 SNMP통보문을 인증하는것이 항상 중요하다. 다른 한편 SNMP통신량을 제기하는것은 필요하지 않을것이다.
5. 인증함수와 기밀성함수의 분리는 기능적유연성을 준다. 실례로 그것은 응용수준에서 인증을 진행하지만 통신량층과 같은 낮은 수준에서 기밀성을 제공하는것이 요구될수 있다.
6. 인증자는 접수시간을 넘어서 보호주기를 늘이며 여전히 통보문내용의 소유를 허용할수도 있다. 통보문암호화에 대하여 통보문을 복호화할 때 보호가 파괴되므로 통보문은 오직 림시적으로만 부정적인 변경에 보호될뿐 목표체계에서는 그렇지 못하다.

마지막으로 MAC는 송수신자들만이 같은 열쇠를 공유하므로 수자서명을 담보하지 못한다는것을 강조한다.

표 8-2는 그림 8-4에서 보여 준 방식의 기밀성함수와 인증함수를 개괄한다.

하쉬함수

통보문인증부호에 대한 변종의 하나는 한방향하쉬함수이다. 통보문인증부호에서처럼 하쉬함수는 가변크기통보문 M을 입력으로 접수하여 출구로서 고정크기하쉬부호 H(M)을 생성해 내는데 때때로 이것을 통보문요약정보라고 한다. 하쉬부호는 통보문비트 모두에 관한 함수로서 오유검출능력을 제공한다. 즉 통보문에서 비트 또는 여러비트에 대한 변경은 하쉬부호에 대한 변경으로 나타난다.

표 8-2. 통보문인증부호 C의 기본리용

ㄱ)A→B: $M \| C_K(M)$ ● 인증을 제공한다. -오직 A와 B만이 K를 공유한다.
ㄴ)A→B: $E_{K_2}[M \| C_{K_1}(M)]$ ● 인증을 제공한다. -오직 A와 B만이 K_1를 공유한다. ●기밀성을 제공한다. -오직 A와 B만이 K_2을 공유한다.
ㄷ) A→B: $E_{K_2}[M] \| C_{K_1}(E_{K_2}[M])$ ● 인증을 제공한다. -K_1를 리용하여 ● 기밀성을 제공한다. -K_2을 리용하여

그림 8-5는 다음과 같이 하쉬부호를 통보문인증을 담보하는데 리용할수 있는 방법의 변종들을 보여 준다.

ㄱ) 통보문+련접된 하쉬부호는 전통암호를 리용하여 암호화된다. 이것은 그림 8-2의 ㄱ에서 보여 준 내부오유조종전략들에 대한 구조와 동일하다. 같은 추리선이 적용된다. 즉 A와 B만이 비밀열쇠를 공유하므로 통보문은 A로부터 와야 하며 변경되지 않는다. 하쉬부호는 인증을 얻는데 요구된 구조 또는 파잉성을 제공한다. 암호화를 완전한 통보문+하쉬부호에 적용하므로 역시 기밀성을 담보한다.

ㄴ) 전통암호를 리용하여 오직 하쉬부호만을 암호화한다. 이것은 기밀성을 요구하지 않는 이런 응용들에 대한 처리부담을 줄인다. 하쉬와 암호화의 결합은 사실상 MAC인 전체적인 함수로 나타난다(그림 8-4의 ㄱ). 즉 $E_K[H(M)]$은 가변길이통보문 M과 비밀열쇠 K에 관한 함수이므로 비밀열쇠를 모르는 적에 대해서는 안전한 고정길이출력을 생성해 낸다.

ㄷ) 공개열쇠암호와 송신자의 비밀열쇠를 리용하여 오직 하쉬부호만을 암호화한다. 항목 ㄴ에서처럼 이것은 인증을 제공한다. 또한 송신자만이 암호화된 하쉬부호를 생성해 낼수 있으므로 수자담보를 제공한다. 사실 이것은 수자서명기술의 본질이다.

ㄹ) 수자서명은 물론 기밀성을 요구한다면 통보문+공개열쇠로 암호화된 하쉬부호는 전통적인 비밀열쇠를 리용하여 암호화될수 있다.

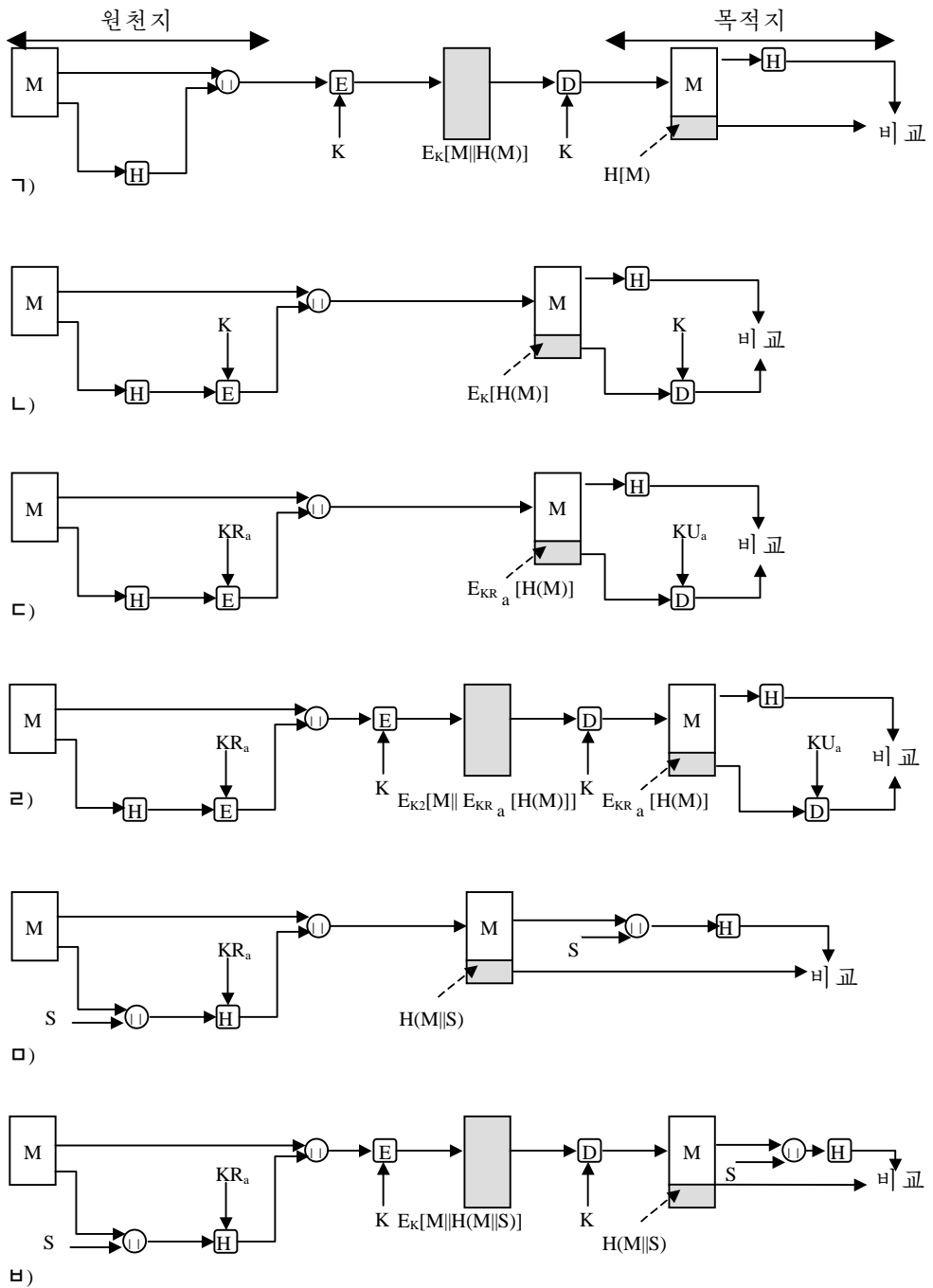



그림 8-5. 하쉬함수의 기본내용

ㅁ) 이 기술은 하쉬함수를 리용하지만 통보문인증을 위한 암호화는 하지 않는다. 이 기술은 두 통신부분이 공통의 비밀값 S를 공유하는것을 가정한다. A는 M과 S의 련접으로 하쉬값을 계산하며 결과하쉬값을 M에 첨부한다. B는 S를 소유하므로

하쉬값을 재계산하여 검증할수 있다. 비밀값자체는 보내지 않으므로 적은 가로챈 통보문을 변경할수 없으며 거짓통보문을 생성할수 없다.

ㅂ) 완전한 통보문+하쉬부호를 암호화함으로써 기밀성을 항목 ㅁ의 방식에 첨가할 수 있다.

기밀성을 요구하지 않을 때 방법 항목 ㄴ과 ㄷ은 보다 적은 계산을 요구하는 완전한 통보문을 암호화하는데서는 우점이 있다. 그럼에도 불구하고 암호화를 피하는 기술에 대한 관심이 더 높아 지고 있다(그림 8-5의 ㅁ). 이에 대한 몇가지 리유를 [TSUD92]에서는 다음과 같이 강조하였다.

- 암호화쏘프트웨어는 속도가 매우 뜨다. 통보문당 암호화될 자료량이 적다 할지라도 체계의 안팎에서 확고한 통보문의 흐름이 있을수 있다.
- 암호화하드웨어비용이 무시될수 없다. DES의 비용이 적게 드는 소편실현이 가능하지만 망의 모든 말단들이 이 능력을 가져야 한다면 비용이 증가한다.
- 암호화하드웨어는 커다란 자료크기에 대하여 최량화된다. 작은 자료블로크에 대하여 높은 시간몫은 초기화/법률화(invocation)로서 보내여 진다.
- 암호화알고리듬은 특권으로써 숨기여 질수 있다. 공개열쇠알고리듬과 같은 일부 암호화알고리듬은 특허를 받아 비용을 증가시켜 승인을 받아야 한다.
- 암호화알고리듬은 미국수출조종위원회의 주요대상이다.

표 8-3은 그림 8-5에서 보여 준 방식의 기밀성과 인증따름을 개괄한다.

8.3 통보문인증부호

암호학적검사합으로 알려 진 MAC는

$$MAC=C_K(M)$$

표 8-3. 하쉬함수 H의 기본리용

ㄱ) A→B: $E_K[M \| H(M)]$ • 기밀성을 제공한다. -A와 B만이 K를 공유한다. • 인증을 제공한다. -H(M)은 암호학적으로 보호된다.	ㄹ) A→B: $E_K[M \| E_{KR_a}[H(M)]$ • 인증과 수자서명을 제공한다. • 기밀성을 제공한다. -A와 B만이 K를 공유한다.
ㄴ) A→B: $M \| E_K[H(M)]$ • 인증을 제공한다. -H(M)은 암호학적으로 보호된다.	ㅁ) A→B: $M \| H(M \| S)$ • 인증을 제공한다. -A와 B만이 S를 공유한다.
ㄷ) A→B: $M \| E_{KR_a}[H(M)]$ • 인증과 수자서명을 제공한다. -H(M)은 암호학적으로 보호된다. -A만이 $E_{KR_a}[H(M)]$을 창조할 수 있다.	ㅂ) A→B: $E_K[M \| H(M) \| S]$ • 인증을 제공한다. -A와 B만이 S를 공유한다. • 기밀성을 제공한다. -A와 B만이 K를 공유한다.

형태의 함수 C에 의해 생성된다. 여기서 M은 가변길이통보문이며 K는 오직 송수신자만이 공유한 비밀열쇠, $C_K(M)$은 고정길이인증자이다. MAC는 통보문이 정확하다는것을 가정하거나 알고 있을 때 어떤 시각에 원천지에서 통보문에 첨부된다. 수신자는 MAC를 재계산함으로써 이 통보문을 인증한다.

이 절에서는 함수 C에 대한 요구를 개괄한 다음에 특수한 실례를 준다. 다른 하나의 실례는 제9장에서 론의한다.

MAC에 대한 요구

대칭 또는 비대칭암호를 리용하여 기밀성을 위해 완전한 통보문을 암호화할 때 이 방식의 보안은 열쇠의 비트길이에 관계된다. 알고리듬의 일부 약점을 제외하고는 적은 가능한 모든 열쇠를 리용하여 힘내기공격에 의거해야 한다. 평균적으로 이런 공격은 k-bit열쇠에 대하여 $2^{(k-1)}$번의 시도를 요구할것이다. 특히 암호문전용공격에 대하여 암호문 C를 가진 적은 모든 가능한 열쇠값 K_i에 대하여 $P_i=D_{K_i}(C)$를 진행하되 P_i가 접수가능한 평문의 형태를 정합하는것을 생성할 때까지 한다.

MAC의 경우에 이런 고찰은 완전히 다르다. 일반적으로 MAC함수는 다값함수이다. 함수의 정의역은 어떤 임의의 길이의 통보문으로 이루어 지지만 값구역은 가능한 모든 MAC와 가능한 모든 열쇠로 이루어 진다. n bit MAC를 리용하면 2^n개의 가능한 MAC가 있지만 $N >> 2^n$인 N개의 가능한 통보문이 있다. 더 나아가서 k bit열쇠에 대하여 2^k개의 가능한 열쇠가 있다.

폭력방법을 리용한다면 적은 열쇠를 발견하기 위해 어떻게 시도할것인가? 기밀성을 채용하지 않는다면 적은 평문통보문과 그에 대응된 MAC를 참조한다. k>n이라고 가정하자. 즉 열쇠크기가 MAC크기보다 크다고 가정하자. 이때 M_1와 MAC_1를 알고 있으면 $MAC_1=C_{K_1}(M_1)$에 대하여 암호분석자는 가능한 모든 열쇠값 K_i에 대하여 $MAC_i=C_{K_i}(M_1)$를 진행할수 있다. $MAC_i=MAC_1$가 정합(대조)되는것을 생성해 내는 적어도 하나의 열쇠가 있다. 총 2^k개의 MAC가 생성되지만 오직 $2^n< 2^k$개의 서로 다른 MAC값들만이 있다. 이리하여 일련의 열쇠들은 정확한 MAC를 생성해 내며 적은 정확한 열쇠를 아는 방법을 가지지 못한다. 평균적으로 총 $2^k/2^n=2^{(k-n)}$개의 열쇠가 정합을 생성한다. 이리하여 적은 다음과 같은 공격을 반복해야 한다.

- **1회전**
  - M_1, $MAC_1=C_K(M_1)$가 주어 졌다.
  - $MAC_i=C_{K_i}(M_1)$를 2^k개의 열쇠모두에 대하여 계산한다.
  - 정합된 수 ≈ $2^{(k-n)}$
- **2회전**
  - M_2, $MAC_2=C_K(M_2)$이 주어 졌다.
  - $MAC_i=C_{K_i}(M_2)$을 나머지 $2^{(k-n)}$개의 열쇠에 대하여 계산한다.
  - 정합된 수 ≈ $2^{(k-2\times n)}$

평균적으로 $k = \alpha \times n$이면 α번의 회전이 요구될것이다. 실례로 80bit열쇠를 리용하여 MAC가 32bit이면 1회전은 대략 2^{48}개의 가능한 열쇠들을 생성해 낼것이다. 2회전은

가능한 열쇠를 대략 2^{16}개의 가능성으로 좁힐것이다. 3회전은 송신자에 의해 리용된 대상이여야 하는 오직 하나의 열쇠만을 생성해 낼것이다.

열쇠길이가 MAC길이를 넘지 않는다면 1회전은 유일한 정합을 생성해 내는것이 적합하다. 한개이상의 열쇠들이 이런 정합을 생성하는것이 가능한데 적의 경우에는 새로운 쌍(통보문, MAC)우에서 같은 검사를 진행할것을 요구한다.

이리하여 인증열쇠를 발견하려는 폭력시도는 효과가 없는것이 아니며 따라서 같은 길이의 복호화열쇠를 발견하는데 요구되는것보다 더 효과적일수 있다. 그러나 열쇠의 발견을 요구하지 않는 다른 공격이 가능하다.

다음과 같은 MAC알고리듬을 고찰하자. $M = (X_1 \| X_2 \| \cdots \| X_m)$을 64bit블로크들의 련접으로 취급하는 통보문이라고 하자. 이때

$$\Delta(M) = X_1 \oplus X_2 \oplus \cdots \oplus X_m$$
$$C_K(M) = E_K[\Delta(M)]$$

을 정의한다. 여기서 ⊕는 배타적론리합(XOR)연산이며 암호화알고리듬은 전자적인 부호책방식의 DES이다. 이리하여 열쇠길이는 56bit이며 MAC길이는 64bit이다. 적이 $\{M \| C_K(M)\}$을 관찰하면 K를 결정하기 위한 폭력시도는 적어도 2^{56}번의 암호화를 요구할것이다. 그러나 적은 X_1부터 X_{m-1}까지를 임의의 요구된 값 Y_1로부터 Y_{m-1}까지로 교체하고 X_m을 Y_m으로 교체함으로써 체계를 공격할수 있다. 여기서 Y_m은 다음과 같이 계산된다.

$$Y_m = Y_1 \oplus Y_2 \oplus \cdots \oplus Y_{m-1} \oplus \Delta(M)$$

적은 이제는 Y_1로부터 Y_m과 수신자에 의해 인증된것으로 접수될 통보문을 형성하는 원래의 MAC로 구성되는 새로운 통보문을 련접할수 있다. 이 교묘한 수법에 대하여 길이 $64 \times (m-1)$bit인 임의의 통보문을 부정적으로 삽입할수 있다.

이리하여 MAC함수의 보안을 평가하는데서 우리에게는 그에 도전할수 있는 공격류형을 고찰하는것이 필요하다. 적이 MAC함수는 알지만 K를 모른다고 가정하자. 이때 MAC함수는 다음과 같은 성질들을 가질것이다.

1. 적이 M과 $C_K(M)$을 관찰한다면 적이 $C_K(M') = C_K(M)$인 통보문 M'를 구성하는것은 계산량적으로 불가능하다.
2. $C_K(M)$은 우연적으로 선택된 통보문 M'와 M에 대하여 $C_K(M') = C_K(M)$일 확률은 2^{-n}이라는 관점에서 평등하게 분포될것이다. 여기서 n은 MAC에서의 비트수이다.
3. M'가 M우에서 어떤 알려 진 변환과 같다고 하자. 즉 $M' = f(M)$이다. 실례로 f는 하나이상의 특수한 비트를 전환할수도 있다. 그 경우에

$$\Pr[C_K(M) = C_K(M')] = 2^{-n}.$$

첫 요구는 적이 비록 열쇠를 모르며 배우려고 하지 않는다고 할지라도 적이 주어 진 MAC를 정합하는 새로운 통보문을 구성할수 있는 앞의 실례를 의미하는것이다. 요구 ②는 선택평균에 기초한 힘내기공격을 좌절시킬 필요성을 취급한다. 즉 적은 K를 모르지

만 MAC함수를 참조하며 MAC생성을 위한 통보문을 표현할수 있다면 적은 주어 진 MAC를 정합하는 대상을 발견할 때까지 여러가지 통보문을 시도할것이다. 함수가 평등분포한다면 폭력방법은 주어 진 MAC와 정합되는 통보문을 발견하기전에는 평균적으로 $2^{(n-1)}$번의 시도를 요구할것이다.

최종요구는 인증알고리듬이 일정한 부분 또는 다른것이 아니라 통보문의 비트들에 관하여 약하지 않을것을 지시한다. 그렇지 않다면 M과 $C_K(M)$을 가진 적은 낡은 MAC를 정합한 새로운 통보문을 보다 빨리 생성해 낼 가능성을 가진 알려 진 《약점》으로 M에 대한 변종을 시도할수 있을것이다.

DES에 기초한 통보문인증부호

일명 자료인증알고리듬이라고 하는 가장 널리 리용되는 MAC들중의 하나는 DES에 기초한것이다. 이 알고리듬은 FIPS발표(FIPS PUB 113)와 ANSI표준(X9. 17)이다.

령초기화벡토르를 가지는 DES연산의 암호문블로크련쇄(CBC)를 리용함으로써 이 알고리듬을 정의할수 있다. 인증될 자료(즉 통보문, 레부호, 파일 또는 프로그람)를 린접한 64bit블로크 D_1, D_2, …, D_N으로 분류한다. 필요하다면 최종블로크를 완전한 64bit블로크를 형성하는 령으로 오른쪽을 메운다. DES암호화알고리듬 E와 비밀열쇠 K를 리용하면 자료인증부호(DAC)를 다음과 같이 계산한다(그림 8-6).

$$O_1 = E_K(D_1)$$
$$O_2 = E_K(D_2 \oplus O_1)$$
$$O_3 = E_K(D_3 \oplus O_2)$$
$$\vdots$$
$$O_N = E_K(D_N \oplus O_{N-1})$$

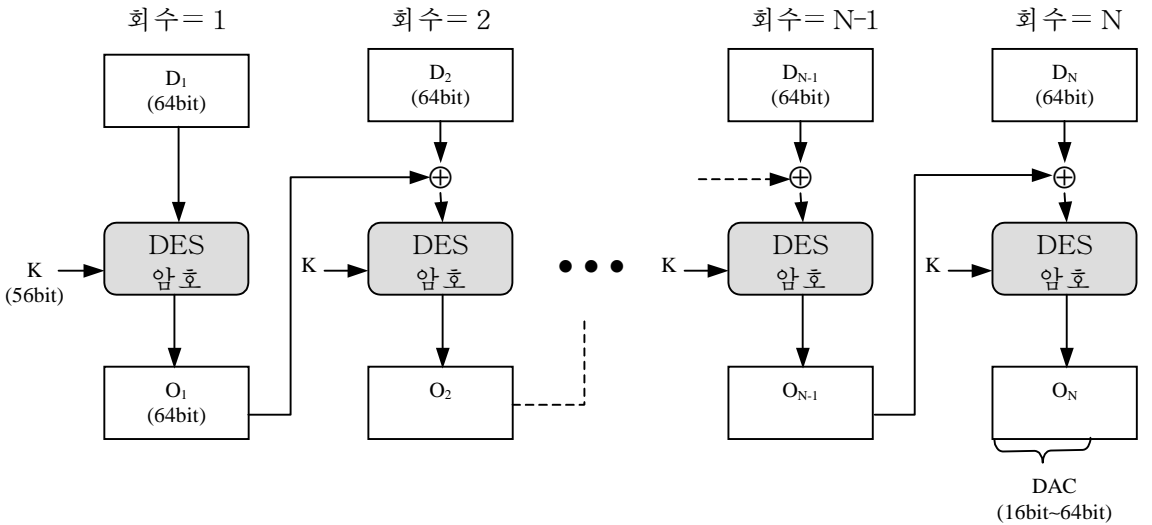



그림 8-6. 자료인증알고리듬(FIPS PUB 113)

DAC는 완전한 블로크 O_N 또는 $16 \le M \le 64$인 블로크의 제일 왼쪽 M bit로 이루어 진다.

이 알고리듬은 앞에서 묘사한 요구들을 만족시킨다.

8.4 하쉬함수

하쉬값은

$$h = H(M)$$

형태의 함수 H에 의해 생성된다. 여기서 M은 가변길이통보문이고 $H(M)$은 고정길이하쉬값이다. 통보문이 정확하다는것을 가정하거나 알 때 원천지에서 하쉬값을 통보문에 첨가한다. 수신자는 하쉬값을 재계산하는것으로써 그 통보문을 인증한다. 하쉬함수자체는 안전하다고 고찰하지 않으므로 하쉬값을 보호하기 위해서는 어떤 수단이 요구된다(그림 8-5).

통보문인증에 쓰이는 하쉬함수에 대한 요구를 설명하는것부터 시작한다. 하쉬함수는 전형적으로 아주 복잡하므로 포함된 론의되는 문제점들을 얻기 위하여 아주 단순한 하쉬함수를 설명하는것이 유용하다. 그다음에는 하쉬함수설계에 대한 여러가지 방식을 본다.

하쉬함수에 대한 요구

하쉬함수의 목적은 파일, 통보문 또는 다른 블로크자료의 《지문》을 생성해 내는것이다. 통보문인증에 쓸모 있게 하기 위하여서는 하쉬함수 H가 다음의 성질들을 가져야 한다[NECH92].

1. H를 임의의 크기의 자료블로크에 적용할수 있다.
2. H는 고정길이출력을 생성해 낸다.
3. 하드웨어와 쏘프트웨어실현이 둘 다 실천적이도록끔하여 임의의 주어 진 x에 대하여 $H(x)$를 계산하는것이 상대적으로 쉽다.
4. 임의의 주어 진 부호 h에 대하여 $H(x)=h$인 x를 구하는것은 계산량적으로 불가능하다. 이것을 때때로 학문적으로는 **한방향성**이라고 한다.
5. 임의의 주어 진 블로크 x에 대하여 $H(y)=H(x)$인 $y \neq x$를 구하는것은 계산량적으로 불가능하다. 이것을 때때로 **약충돌방지**라고 한다.
6. $H(x)=H(y)$인 임의의 쌍 (x, y)를 구하는것은 계산량적으로 불가능하다. 이것을 때때로 **강충돌방지**라고 한다(불행하게도 이런 용어들은 모순없이는 리용되지 못한다. 학문적으로 리용된 다른 하나의 용어들은 한방향성하쉬함수(성질 4와 6), 충돌방지하쉬함수(성질 4, 5 및 6), 약한방향하쉬함수(성질 4와 5), 강한방향하쉬함수(성질 4, 5 및 6)이다. 독자들은 쓰이는 용어의 의미를 결정하는 학문을 읽는데 각별한 주의를 돌려야 한다).

첫 3개 성질들은 통보문인증에 하쉬함수를 실천적으로 응용하기 위한 요구이다.

네번째 성질은 한방향성이다. 통보문이 주어 졌을 때 부호를 생성하는것은 쉽지만

부호가 주어 졌을 때 통보문을 생성하는것은 실제상 불가능하다. 인증기술이 비밀값을 리용한다면 이 성질은 중요하다(그림 8-5의 ㅁ). 비밀값자체를 보내지는 않지만 하쉬함수가 한방향이 아니라면 공격자는 쉽게 비밀값을 발견할수 있다. 공격자가 전송을 관찰하거나 가로챌수 있다면 공격자는 통보문 M과 하쉬부호 $C=H(S_{AB} \| M)$을 얻는다. 그다음에 공격자는 하쉬함수를 거꾸로 전환하여$S_{AB} \| M=H^{-1}(C)$를 얻는다. 이제는 공격자가 M과 $S_{AB} \| M$을 둘 다 가지므로 S_{AB}를 발견하는것은 아주 쉬운 문제이다.

다섯번째 성질은 같은 값을 주어 진 통보문으로 하쉬하는 다른 하나의 통보문을 구할수 없다는것을 강조한다. 이것은 암호화된 하쉬부호를 리용할 때 위조를 방지한다(그림 8-5의 ㄴ과 ㄷ). 이 경우에 대하여 적은 통보문을 읽으므로 자기의 하쉬부호를 생성할수 있다. 그러나 적은 비밀열쇠가 없으므로 검출없이는 통보문을 변경할수 없다. 이 성질이 성립하지 않는다면 공격자는 다음과 같은 렬을 만들어 낼수 있다. 우선 통보문+그의 암호화된 하쉬부호를 관찰하거나 가로채여 통보문으로부터 이 암호화된 하쉬부호를 생성하며 다음으로는 같은 하쉬부호를 가지는 다른 하나의 통보문을 생성한다.

여섯번째 성질은 하쉬함수가 간단히 설명하는 생일공격클라스에서 어떻게 방지되는가에 귀착된다.

단순하쉬함수

모든 하쉬함수들은 다음과 같은 일반원리를 리용하여 조작한다. 입력(통보문, 파일 등)을 n bit블로크들의 렬로 고찰한다. 입력은 어떤 시각에 n bit하쉬함수를 반복적인 방식으로 하나의 블로크씩 처리해 나간다.

가장 단순한 하쉬함수들중의 하나는 매 블로크의 비트별 배타적론리합(XOR)이다. 이것을 다음과 같이 표현할수 있다.

$$C_i = b_{i1} \oplus b_{i2} \oplus \cdots \oplus b_{im}$$

여기서

C_i=하쉬부호의 i번째 비트, $1 \le i \le n$
m=입력에서 $n$bit블로크의 수
b_{ij}=j번째 블로크에서 i번째 비트
$\oplus$=XOR연산

그림 8-7은 이 연산을 보여 준다. 즉 그것은 매 비트위치에 대하여 단순한 기우성을 생성해 내며 경도의 파잉성검사로 알려 져 있다. 이것은 자료완정성검사로서 우연자료에 대하여 응당히 효과적이다. 매 $n$bit하쉬값은 등식적으로 적합하다. 이리하여 자료오유가 변경되지 않은 하쉬값을 나타낼 확률은 2^{-n}이다. 보다 예언적으로 형식화된 자료에 대하여서는 이 함수가 효과성이 적다. 실례로 가장 표준적인 본문파일에서 매 옥테드(octet)의 높은 위치비트는 항상 령이다. 그래서 128bit하쉬값을 리용하면 2^{-128}이라는 효과성대신에 이런 류형의 자료에 대한 하쉬함수의 효과성은 2^{-112}이다.

문제를 해결하는 단순한 방도는 매 블로크를 처리한 다음에 하쉬값에 대하여 한 비트순환밀기 또는 회전을 진행하는것이다. 이 절차를 다음과 같이 개괄할수 있다.

	비트 1	비트 2	…	비트 n
블로크 1	b_{11}	b_{21}		b_{n1}
블로크 2	b_{12}	b_{22}		b_{n2}
	⋮	⋮	⋮	⋮
블로크 m	b_{1m}	b_{2m}		b_{nm}
하쉬부호	C_1	C_2		C_n

그림 8-7. 비트별 XOR를 리용한 단순하쉬함수

1. 초기에 $n$bit하쉬값을 령으로 설정한다.
2. 매 련속적인 $n$bit자료블로크를 다음과 같이 처리한다.
   ㄱ) 현재하쉬값을 왼쪽으로 한비트 회전시킨다.
   ㄴ) 블로크를 하쉬값과 배타적론리합한다.

이것은 입력을 더 완전하게 《우연화》하며 입력에 나타나는 임의의 규칙성을 극복하는 효과를 가진다. 그림 8-8은 16bit하쉬값에 대한 이런 두가지 류형의 하쉬함수를 보여 준다.

두번째 절차는 비록 자료완정성이 좋은 척도를 제공한다고 할지라도 그것은 그림 8-5의 ㄴ과 ㄷ에서 보여 준바와 같이 암호화된 하쉬부호를 평문통보문과 함께 리용할 때 자료보안을 위하여 실제적으로는 리용하지 못한다. 통보문이 주어 졌을 때 그 하쉬부호를 산생하는 새로운 통보문을 생성해 내는것은 쉬운 문제이다. 단순히 요구되는 다른 하나의 통보문을 준비한 다음에 새로운 통보문+요구되는 하쉬부호를 산생하는 블로크에 집중되는 n bit블로크를 첨가한다.

오직 하쉬값만이 암호화되였다면 비록 단순한 XOR나 회전 XOR(RXOR)이 불충분하다고 할지라도 아직 하쉬부호는 물론 통보문을 암호화할 때 이런 단순한 하쉬함수가 유용하다는것을 느낄수 있을것이다(그림 8-5의 ㄱ). 그러나 주의해야 한다. NBS(민족표준국)에 의하여 최초에 제안된 기술은 단순한 XOR를 리용하여 64bit통보문블로크에 적용하였으며 그다음에는 암호문블로크련쇄(CBC)방식을 리용한 완전한 통보문의 암호화에 적용하였다. 이 방식을 다음과 같이 정의할수 있다. 64bit블로크렬 X_1, X_2, …, X_N으로 이루어 진 통보문이 주어 졌을 때 블로크별 XOR 또는 모든 블로크로서 하쉬부호 C를 정의하고 하쉬부호를 최종블로크로서 첨부한다. 즉

$$C = X_{N+1} = X_1 \oplus X_2 \oplus \cdots \oplus X_N$$

다음으로 암호화된 통보문 Y_1, Y_2, …, Y_{N+1}를 생성해 내는 CBC방식을 리용하여 완전한 통보문+하쉬부호를 암호화한다. [JUEN85]는 이 통보문의 암호문을 하쉬부호에 의해서는 검출불가능한 방식으로 처리할수 있는 몇가지 방식을 강조하였다. 실례로 CBC의 정의(그림 3-12)에 의하여

$$X_1 = IV \oplus D_K(Y_1)$$
$$X_i = Y_{i-1} \oplus D_K(Y_i)$$
$$X_{N+1} = Y_N \oplus D_K(Y_{N+1})$$

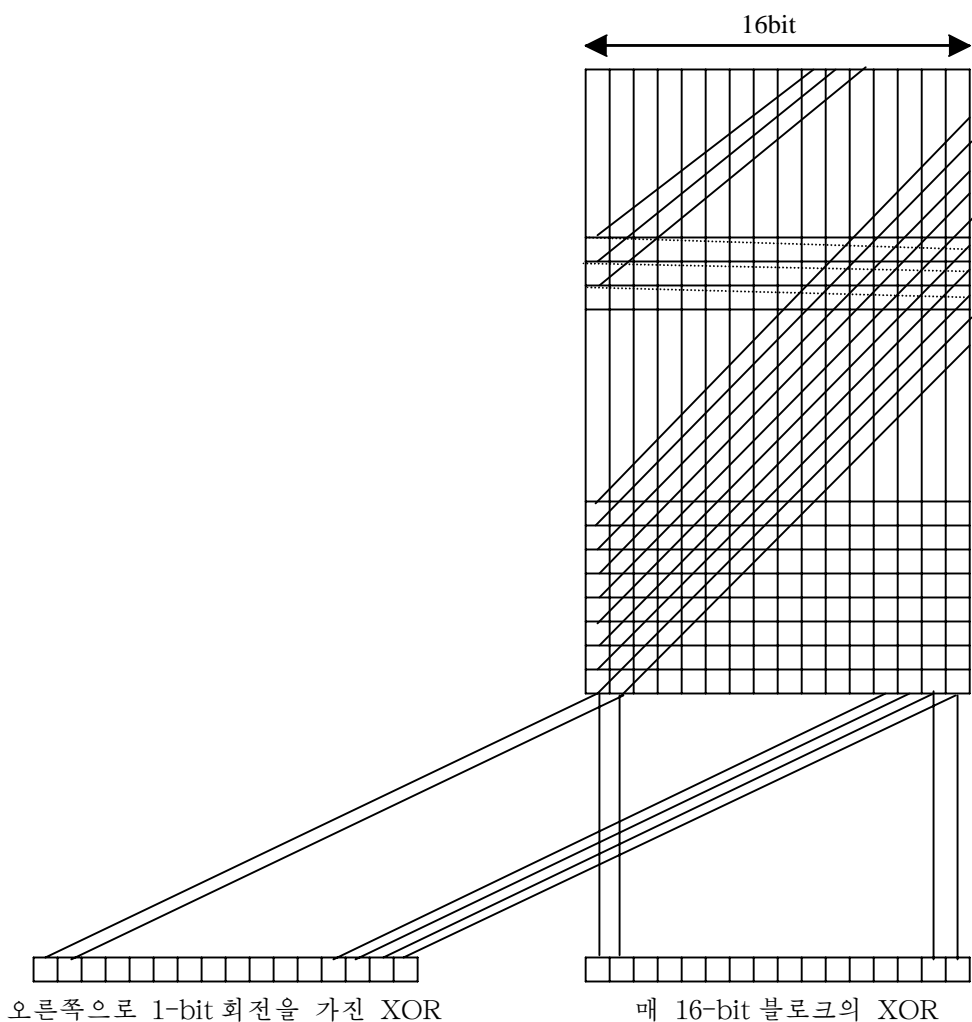



그림 8-8. 두개의 단순하쉬함수

이지만 X_{N+1}는 다음과 같은 하쉬부호이다.

$$X_{N+1} = X_1 \oplus X_2 \oplus \cdots \oplus X_N$$
$$= (IV \oplus D_K(Y_1)) \oplus (Y_1 \oplus D_K(Y_2)) \oplus \cdots \oplus (Y_{N-1} \oplus D_K(Y_N))$$

선행한 식의 항들은 임의의 순서로 배타적론리합을 할수 있으므로 암호문블로크가 치환되였다면 하쉬부호를 변경시킬수 없다.

생일공격

64bit하쉬부호를 리용한다고 가정하자. 이것은 아주 안전하다고 생각할수 있을것이다. 실례로 암호화된 하쉬부호 C를 대응하는 암호화되지 않은 통보문 *M*과 함께 전송한

다면(그림 8-5의 ㄴ과 ㄷ) 적은 다른 통보문을 치환하여 수신자를 속이는 H(M)=H(M')인 M'를 구하는것이 필요하다. 평균적으로 적은 가로챈 통보문의 하쉬부호와 정합되는것을 구하기 위하여 대략 2^{63}개의 통보문을 시도해야 할것이다(부록 8의 식 8-1을 보시오).

그러나 생일모순에 기초하여 다른 종류의 공격이 가능하다(부록 8). 유발 (Yuval)은 다음과 같은 전략을 제기하였다[YUVA79].

1. 원천지 A는 적당한 m bit MAC를 첨부하고 그 MAC와 A의 비밀열쇠를 암호화함으로써 통보문《서명》을 준비한다.
2. 적은 본질적으로 같은 의미를 나르는 통보문모두에 대한 $2^{m/2}$개의 변종을 생성한다. 적은 실제대상을 위하여 치환하는 부정적인 통보문에 대한 변종인 통보문모두의 같은 수를 준비한다.
3. 같은 하쉬부호를 생성해 내는 통보문 쌍을 구하기 위하여 두 통보문모임을 비교한다. 생일모순에 의한 성공확률은 0.5보다 크다. 정합되지 않는다면 보충적인 타당한 통보문과 부정적인 통보문을 정합할 때까지 생성한다.
4. 적은 서명을 위하여 A에 대한 타당한 변종을 제공한다. 그다음에 이 서명을 지향된 수신자에게 전송할 부정적인 변종에 붙일수 있다. 두 변종이 같은 하쉬부호를 가지므로 같은 서명을 생성해 낼것이며 적은 암호화열쇠를 비록 모른다고 할지라도 성공을 담보한다.

이리하여 64bit하쉬부호가 리용되면 요구된 효과수준은 오직 2^{32}의 정도이다(부록 8과 식 8-7을 보시오).

같은 의미를 나르는 많은 변종의 생성은 힘들지 않다. 실례로 적은 문서전반에 걸쳐 단어들사이의 일련의 《공백-공백-역행(space-space-backspace)》문자쌍들을 삽입할수 있을것이다. 그다음에 선택된 실체로 《공백-역행-공백(space-backspace-space)》을 치환함으로써 변종을 생성할수 있을것이다.

또 다른 한가지 방식으로서 적은 통보문을 단순히 바꾸어 말하지만 의미를 귀환할수 있을것이다. 그림 8-9는 실례를 하나 보여 준다[DAVI89].

이로부터 얻어 진 결론은 하쉬부호의 길이가 본질적이라는것이다. 이에 대해서는 8.5에서 더 론의할것이다.

블로크련쇄기술

암호문블로크련쇄기술을 리용하지만 비밀열쇠가 요구하지 않는 하쉬함수를 제기하는 일련의 시도들이 있다. 이러한 첫 제기중의 하나가 라빈(Rabin)의 문헌[RABI78]이다. 통보문 M을 고정길이블로크 M_1, M_2, …, M_N으로 나누고 DES와 같은 전통암호를 리용하여 다음과 같이 하쉬부호 G를 계산한다.

$$H_0 = 초기값$$
$$H_i = E_{M_i}[H_{i-1}]$$
$$G = H_N$$

Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / – –} Alfred {P. / – –} Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the} Northern {European / Europe} {area / division}. He {will take / has taken} over {the / – –} responsibility for {all / the whole of} our interest in {watches and jewellery / jewellery and watches} in the {area / region}. Please {afford / give} him {every / all the} help he {may need / needs} to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the {latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum} of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document} as proof of identity. An order with his signature, which is {appended / attached} {authorizes / allows} you to charge the cost to this commany at the {above / head office} address. We {fully / – –} expect that our {level / volume} of orders will increase in the {following / next} year and {trust / hope} that the new appointment will {be / prove} {advantageous / an advantage} to both our companies.

그림 8-9. 2^{37}개 방안의 편지 [DAVI89]

이것은 CBC기술과 류사하지만 이 경우에는 비밀열쇠가 없다. 임의의 하쉬부호에서처럼 이 방식은 생일공격에 대한 학문이며 그리고 암호화알고리듬이 DES이고 오직 64bit하쉬부호만을 생성해 낸다면 체계는 견고하지 못하다.

더 나아가서 생일공격의 다른 판은 적이 오직 하나의 통보문과 그의 타당한 서명만을 참조하며 다중서명을 얻을수 없는 경우에도 리용될수 있다. 씨나리오는 다음과 같다. 적이 암호화된 하쉬부호의 형태로 서명과 통보문을 가로채고 암호화되지 않은 하쉬부호는 mbit이라고 가정한다.

1. 이 항목의 시작에서 정의된 알고리듬을 리용하여 암호화되지 않은 하쉬부호 G를 계산한다.
2. 임의의 요구된 통보문을 Q_1, Q_2, …, Q_{N-2}형태로 구성한다.
3. $1 \le i \le (N-2)$에 대하여 $H_i = E_{Q_i}[H_{i-1}]$를 계산한다.
4. $2^{m/2}$개의 우연블로크를 생성하고 매 블로크 X에 대하여 $E_X[H_{N-2}]$을 계산한다. 보충적인$2^{m/2}$개의 블로크를 생성하고 매 블로크 Y에 대하여 $D_Y[G]$를 계산한다. 여기서 D는 E에 대응하는 복호화함수이다.
5. 생일모순에 기초하여 높은 확률을 가진 $E_X[H_{N-2}] = D_Y[G]$인 X와 Y가 있을것이다.
6. 통보문 Q_1, Q_2, …, Q_{N-2}, X, Y를 형성한다. 이 통보문은 하쉬부호 G를 가지므로 가로챈 암호화된 서명과 함께 리용될수 있다.

이런 형태의 공격은 《중간에서 만나다》로 알려 져 있다. 일련의 연구자들은 기본블로크련쇄방식을 강화하는데 지향된 세분을 제기하였다. 실례로 다비에즈(Davies)와 프라이스(Price)[DAVI89]는 다음과 같은 방안을 서술하였다.

$$H_i = E_{M_i}[H_{i-1}] \oplus H_{i-1}$$

[MEYE88]에서 제기된 다른 하나의 변종은 다음과 같다.

$$H_i = E_{M_{i-1}}[M_i] \oplus M_i$$

그러나 이 방식들은 둘 다 공격의 변종[MIYA90]에 약하다는것을 보여 준다. 더 일반적으로 어떤 형태의 생일공격은 결과하쉬함수가 충분히 작거나(즉 64bit이하) 보다 큰 하쉬함수를 독립적인 부분부호들로 분해할수 있다면 비밀열쇠없이 암호문블로크련쇄를 리용하여 임의의 하쉬방식에 대하여 계속될수 있다는것을 보여 줄수 있다[JUEN87].

이리하여 하쉬하는 다른 방식을 구하는데 주의를 돌리게 된다. 이런것들의 대부분은 약점을 가진다는것을 보여 주었다[MITC92]. 9장에서 보다 강한 몇가지 하쉬함수들을 설명한다.

8.5 하쉬함수와 MAC의 보안

공개열쇠암호와 전통암호에서처럼 하쉬함수와 MAC에 대한 공격을 2개 부류 즉 힘내기공격과 암호분석으로 분류할수 있다.

힘내기공격

힘내기공격의 본질은 하쉬함수와 MAC와는 좀 차이난다.

하쉬함수

힘내기공격에 대한 하쉬함수의 강도는 알고리듬에 의해 생성된 하쉬부호의 길이에만 관계된다. 다음과 같은 3개의 가능한 성질들이 있는 하쉬함수에 대한 우리의 론의를 상기하자.

- **한방향:** 임의의 주어 진 부호 h에 대하여 $\mathrm{H}(x)=h$인 x를 구하는것은 계산량적으로 불가능하다.
- **약충돌방지:** 임의의 주어 진 블로크 x에 대하여 $\mathrm{H}(y)=\mathrm{H}(x)$인 $y\neq x$를 구하는것은 불가능하다.
- **강충돌방지:** $\mathrm{H}(x)=\mathrm{H}(y)$인 임의의 쌍 (x,y)를 구하는것은 계산량적으로 불가능하다.

길이 n인 부호에 대하여 본바와 같이 요구된 효과수준은 다음과 같은것에 비례한다.

한방향	2^n
약충돌방지	2^n
강충돌방지	$2^{n/2}$

강충돌방지를 요구한다면 (그리고 이것이 일반목적을 위해 안전한 하쉬부호로 가능하다면) 값 $2^{n/2}$은 힘내기공격에 대한 하쉬부호의 강도를 결정한다. 우소트(Oorshot)와 와이너(Wiener)의 문헌[OORS94]는 MD5에 대한 1천만$짜리 충돌탐색기계의 설계를 제기하였는데 이것은 128bit하쉬길이를 가지며 24일간후에야 충돌을 구할수 있었을것이다. 이리하여 128bit부호를 불충분한것으로 평가할수 있다. 다음단계에서는 하쉬부호를 32bit렬로 취급하면 160bit하쉬길이이다. 160bit의 하쉬길이에 대하여 동일한 탐색기계는 충돌을 구하는데 4천년이상을 요구하게 된다. 현재 9장에서 론의되는 두개의 가장 극적인 하쉬부호 SHA-1과 RIPEMD-160은 160bit하쉬부호길이이다.

통보문인증부호

MAC에 대한 힘내기공격은 알려 진 통보문－MAC쌍을 요구하므로 더 힘든 보증이다. 이것이 왜 그런가를 보자. 하쉬부호를 공격하기 위하여 다음과 같은 방식으로 처리할수 있다. 고정된 통보문 x와 $n$bit하쉬부호 $h=\mathrm{H}(x)$가 주어 졌을 때 충돌을 구하는 폭력방법은 우연비트렬 y를 포착하여 $\mathrm{H}(y)=\mathrm{H}(x)$를 검사하는것이다. 공격자는 이것을 반복적으로 그리고 비직결식으로 할수 있다. 비직결공격을 MAC알고리듬에 리용할수 있는가 없는가는 열쇠의 상대적크기와 MAC에 관계된다.

처리를 하기 위하여 다음과 같이 표현할수 있는 MAC알고리듬의 요구된 보안을 규

정하는것이 필요하다.

- **계산방지:** 하나이상의 본문-MAC쌍$(x_i, C_K(x_i))$들이 주어 졌을 때 임의의 새로운 입력 $x \neq x_i$에 대하여 임의의 본문－MAC쌍$(x, C_K(x))$을 계산하는것은 계산량적으로 불가능하다.

다른 말로 하면 공격자는 주어 진 통보문 x에 대하여 타당한 MAC부호에로 도달하였으면 한다. 가능한 공격의 2개의 선이 있다. 열쇠공간의 공격과 MAC값의 공격이다. 이 매개를 실제로 설명하자.

공격자가 MAC열쇠를 결정할수 있으면 임의의 입력 x에 대하여 타당한 MAC값을 생성하는것이 가능하다. 열쇠크기가 $k$bit이고 공격자는 하나의 알려 진 본문－MAC쌍을 가지고 있다고 가정하자. 이때 공격자는 가능한 모든 열쇠에 대하여 알려 진 본문우에서 $n$bit MAC를 계산할수 있다. 적어도 하나의 열쇠가 정확한 MAC 즉 초기에 알려 진 본문-MAC쌍을 생성해 내는데 리용된 타당한 열쇠를 생성해 낸다는것을 담보한다. 공격의 이 단계는 2^k에 비례하는 효과수준을 취한다(즉 2^k개의 가능한 열쇠값의 매개에 대하여 하나의 연산). 그러나 앞에서 서술한바와 같이 MAC는 다값넘기기이므로 정확한 값을 생성해 내는 다른 열쇠가 있을수도 있다. 이리하여 정확한 값을 생성해 내는 1개이상의 열쇠를 발견하려면 보충적인 본문－MAC쌍을 검사해야 한다. 효과수준이 매 보충적인 본문－MAC쌍과 함께 급격히 떨어 지며 효과의 전체 수준은 거칠게 2^k이라는것을 보여줄수 있다[MENE97].

공격자는 또한 열쇠를 발견하기 위한 시도없이 MAC값우에서 동작할수 있다. 여기서 그 대상은 주어 진 MAC값을 정합하는 통보문을 구하거나 주어 진 통보문에 대한 타당한 MAC값을 생성하는것이다. 어느 경우에나 효과수준은 하쉬부호의 한방향 또는 약충돌방지성 또는 2^n를 공격하는것과 호환가능하다. MAC의 경우에 공격은 그 이상의 입력없이는 비직결적으로 유도할수 없으며 공격자는 선택된 본문-MAC쌍 또는 열쇠에 대한 지식을 요구할것이다.

개괄적으로 말하면 MAC알고리듬에 대한 힘내기공격의 효과수준을 $\min(2^k, 2^n)$으로 표현할수 있다. 강도평가는 대칭암호화알고리듬에 대한것과 류사하다. 열쇠길이와 MAC길이는 $\min(k,\ n) \geq N$인 관계를 만족시킬것을 요구하는것이 합리적일것이다. 여기서 N은 아마도 128bit의 령역에 속한다.

암호분석

암호화알고리듬에서처럼 하쉬함수와 MAC에 대한 암호분석공격은 전면탐색이 아닌 일정한 공격을 진행하는 어떤 알고리듬의 성질들을 개척할것을 요구한다. 암호분석에 대한 하쉬나 MAC알고리듬의 방지를 측정하는 방법은 힘내기공격에 요구된 효과와 그 강도를 비교하는것이다. 즉 MAC알고리듬의 기본하쉬는 폭력효과이상인 암호분석적효과를 요구할것이다.

하쉬함수

최근년간에 하쉬함수에 대한 암호분석적공격을 개발하는데서 상당한 효과와 일정한 성과들이 있다. 이것을 리해하기 위하여서는 그림 8-10에서 보여 준 전형적으로 안전한 하쉬함수의 전면적구조를 보는것이 필요하다.

반복하쉬함수라고 하는 이런 구조를 머클러(Merkle)의 문헌[MERK89]가 제안하였는데 이것은 9장에서 묘사하는 MD5, SHA-1 그리고 RIPEMD-10모두를 포괄하는 현재 리용되는 대부분의 하쉬함수의 구조이다. 하쉬함수는 입력통보문을 취하여 그것을 매개가 *b*bit인 *L*-1개의 고정크기블로크로 분할한다. 필요하다면 최종블로크는 *b*bit를 채워넣는다. 최종블로크는 또한 하쉬함수에 대한 입력의 총 길이값을 포함한다. 길이의 포함은 적에게 더 힘든 일감을 주게 한다. 적은 같은 값으로 하쉬하는 같은 길이의 두 통보문을 구해야 하거나 그 길이값들과 함께 같은 값에로 하쉬하는 서로 다른 길이의 두 통보문을 구해야 한다.

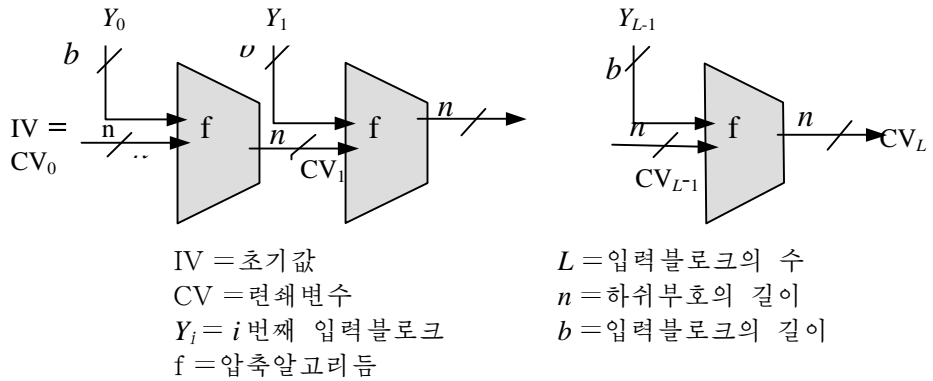



그림 8-10. 안전한 하쉬부호의 일반구조

하쉬알고리듬은 두 입력(련쇄변수라고 하는 선행단계로부터의 *n*bit입력과 *b* bit블로크)을 취하여 *n*bit출력을 생성해 내는 압축함수 f의 반복리용을 포함한다. 하쉬의 시작에서 련쇄변수는 알고리듬의 부분으로 묘사된 초기값을 가진다. 련쇄변수의 최종값은 하쉬값이다. 보통 $b>n$이므로 항압축이다. 하쉬함수를 다음과 같이 개괄할수 있다.

$$CV_0 = IV = \text{초기 } n\text{bit값}$$
$$CV_i = f(CV_{i-1}, Y_{i-1}), \quad 1 \le i \le L$$
$$H(M) = CV_L$$

여기서 하쉬함수에 대한 입력은 블로크 $Y_0, Y_1, \cdots, Y_{L-1}$로 이루어 진 통보문 *M*이다.

이 반복적인 구조의 동기는 압축함수가 충돌방지이면 결과의 반복하쉬함수도 충돌방지이라는 머클러의 문헌[MERK89]와 댐가드(Damgard)의 문헌[DAMG89]로부터 나온다(역은 반드시 참으로 되는것이 아니다). 그러므로 이 구조를 리용하여 임의의 길이의 통보문을 조작하는 안전한 하쉬함수를 생성해 낼수 있다. 안전한 하쉬함수를 설계하는 문제는 어떤 고정된 크기의 입력을 조작하는 충돌방지압축함수를 설계하는 문제에 귀착된다.

하쉬함수의 암호분석은 f의 내부구조에 집중되며 f의 하나의 실행에 대하여 충돌을 생성해 내는 효과적인 기술을 구하려는 시도에 기초하고 있다. 일단 그렇게 되면 공격은 IV의 고정된 값을 고려해야 한다. f에 대한 공격은 그 내부구조를 밝히는데 관계된다.

전형적으로 대칭블로크암호문에서처럼 f는 처리의 회전들의 계렬로 이루어 지므로 공격은 회전에서 회전에로의 비트변경의 패턴을 분석한다.

임의의 하쉬함수에 대하여 적어도 블로크크기 b와 같은 길이의 통보문을 길이 n인 하쉬부호에로 넘기기때문에 충돌이 존재해야 한다는것을 잊지 말자. 여기서 $b<n$이다. 요구되는것은 충돌을 구하는것을 계산량적으로 불가능하게 하는것이다.

하쉬함수로 설치한 공격은 오히려 복잡하며 여기서는 우리의 범위를 벗어 난다. 흥미 있는 독자들은 문헌[DOBB96a]와 [BELL97]를 보시오.

통보문인증부호

하쉬함수보다도 MAC의 구조변경이 더 많이 있으므로 MAC의 암호분석에 대하여 일반화하는것이 힘들다. 멀지 않아 이런 공격을 개발하는것이 완성될것이다. 특수한 MAC를 공격하는 유용한 최초의 문헌은 [PREN96]이다.

참고문헌

문헌[JUEN85]와 [JUEN87]은 암호학적MAC와 하쉬함수에 대해 기본론의하며 통보문인증에 대한 좋은 환경을 제공한다. 하쉬함수와 통보문인증부호에 대한 구체적인 취급은 문헌[STIN95]와 [MENE97]에 있다.

JUEN85 Jueneman, R.; Matyas, S.; and Meyer, C. "Message Authentication." *IEEE Communications Magazine*, September 1988.

JUEN87 Jueneman, R. "Electronic Document Authentication." *IEEE Network Magazine*, April 1987.

MENE97 Menezes, A.; Oorshcot, P.; and Vanstone, S. *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.

STIN95 Stinson, D. Cryptography: *Theory and Practice*. Boca Raton, FL: CRC Press, 1995.

문 제

1. F가 오유검출함수이면 내부 또는 외부의 리용(그림 8-2)은 오유검출능력을 제공할것이다. 전송된 통보문의 임의의 비트가 변경되면 FCS함수가 암호화함수의 아낙 또는 바깥에서 진행되는가에 따라 이것은 수신된 FCS와 계산된 FCS의 불일치(mismatch)로 반영될것이다. 어떤 부호들은 또한 오유교정능력을 제공한다. 함수의 속성에 따라 한비트 또는 작은 비트수가 전송중에 (림시적으로)변경되면 오유교정부호는 충분한 과잉정보를 가지여 오유비트 또는 비트들을 결정하고 그것들을 교정한다. 분명히 오유교정부호는 암호화변수에 외부적으로 리용될 때 오유교정능력을 제공할것이다. 암호화함수에 내부적으로 리용된다면 그것은 역시 이 능력을 제공하겠는가?

2. 8.3에서 묘사된 자료인증알고리듬을 령초기화벡토르를 가진 연산의 암호문블로크련쇄(CBC)방식을 리용하여 정의할수 있다(그림 8-6). 같은 결과를 암호문반결합방식을 리용하여 생성해 낼수 있다는것을 보여 주시오.
3. 고속통신규약 XTP는 두개의 16bit함수 XOR와 RXOR의 련접으로 정의된 32bit검사합함수를 리용한다. 여기서 XOR와 RXOR는 8.4에서 《두개의 단순하쉬함수》로 정의하였으며 그림 8-8로 보여 주었다.
   ㄱ) 이 검사합이 홀수의 오유비트에 의해 생긴 오유모두를 검출하겠는가? 설명하시오.
   ㄴ) 이 검사합이 짝수의 오유비트에 의해 생긴 오유모두를 검출하겠는가? 아니라면 검사합이 실패로 되는 오유패턴을 특징지으시오.
   ㄷ) 인증을 위한 하쉬함수로서 이 함수를 리용할 때 효과성을 설명하시오.
4. ㄱ) 8.4에서 서술한 다비에즈와 프라이스하쉬부호방식을 고찰하여 DES를 다음과 같은 암호화알고리듬으로 리용한다고 가정하자.

$$H_i = E_{M_i}[H_{i-1}] \oplus H_{i-1}$$

그리고 DES의 보충적인 성질(문제 3-10)을 상기하자. 즉 $Y = DES_K(X)$이면 $Y' = DES_{K'}(X')$이다. 이 성질을 리용하여 블로크 M_1, M_2, …, M_N으로 이루어 진 통보문을 그의 하쉬부호를 변경함이 없이 어떻게 변경할수 있는가를 보여 주시오.
   ㄴ) 류사한 공격이 문헌[MEYE88]에서 제기된 다음과 같은 방식에 대해서 성공적이라는것을 보여 주시오.

$$H_i = E_{H_{i-1}}[M_i] \oplus M_i$$

5. 하쉬함수를 리용하여 DES와 류사한 구조를 가진 블로크암호문을 구성하는것이 가능하다. 하쉬함수는 한방향이며 블로크암호화는(복호화와) 가역이여야 하기때문에 그것이 가능한가?
6. 이제는 반대의 문제를 고찰하자. 암호화알고리듬을 리용하여 한방향하쉬함수를 구성한다. 알려 진 열쇠를 가진 RSA를 리용하는것을 고찰하자. 이때 블로크렬로 이루어진 통보문을 다음과 같이 처리한다. 첫번째 블로크를 암호화하고 그 결과와 두번째 블로크를 배타적론리합하여 이 과정을 반복한다. 다음의 문제를 풂으로써 이 방식은 안전하지 못하다는것을 보여 주시오. 두 블로크통보문 B1, B2와 그의 하쉬함수

$$RSAH(B1, B2) = RSA(RSA(B1) \oplus B2)$$

이 주어 졌다. 임의의 블로크 C1이 주어 졌을 때 RSAH(C1, C2) = RSAH(B1, B2)인 C2를 택하시오.

부록 8. 생일공격의 수학적기초

이 부록에서는 생일공격의 수학적정당성을 유도한다. 관련된 문제부터 시작하여 다음 이름 《생일공격》이 유도된 문제를 본다.

관련된 문제

하쉬함수와 관련된 일반문제는 다음과 같다. 하쉬함수가 주어 졌을 때 n개의 가능한 입력과 하나의 특수값 $H(x)$에 대하여 H를 k개의 우연입력에 적용하면 적어도 하나의 입력 y가 $H(y)=H(x)$를 만족시킬 확률이 0.5인 k의 값은 무엇이여야 하는가?

하나의 y값에 대하여 $H(y)=H(x)$일 확률은 바로 $1/n$이다. 반대로 $H(y)\neq H(x)$일 확률은 $[1-(1/n)]$이다. k개의 우연 y값을 생성하면 그들중 어느것도 정합되지 않을 확률은 바로 매 개별적값들이 정합되지 않는 확률들의 적이거나 $[1-(1/n)]^k$이다. 이리하여 적어도 하나를 정합할 확률은 $1-[1-(1/n)]^k$이다.

이제는 이항정리를 다음과 같이 정식화할수 있다.

$$(1-a)^k=1-ka+\frac{k(k-1)}{2!}a^2-\frac{k(k-1)(k-2)}{3!}a^3\ \cdots$$

아주 작은 a값에 대하여 이것은 $(1-ka)$로 근사화할수 있다. 이리하여 적어도 하나를 정합할 확률은 $1-[1-(1/n)]^k\approx 1-[1-(k/n)]=k/n$으로 근사화한다. 0.5의 확률에 대하여 $k=n/2$이다.

특히 $m$bit하쉬부호에 대하여 가능한 하쉬부호는 2^m이고 절반확률을 생성해 낼 k의 값은 다음과 같다.

$$k = 2^{(m-1)} \qquad (8\text{-}1)$$

생일모순

생일모순은 확률결과가 때때로 비직관적이라는것을 보여 주는 초등확률론에서 자주 제기된다. 이 문제를 다음과 같이 정식화할수 있다. 확률이 0.5이상이고 k명집단에서 적어도 2명의 생일이 같은 k의 최소값은 얼마인가? 2월 29일을 무시하고 매 생일이 항등적으로 적당하다고 가정한다. 대답하기 위하여

$P(n,k)=\Pr$[적어도 한명이 k항목으로 중복되며 매 항목에서 1과 n사이의 n개의 항등적으로 적당한 값들중의 하나를 취할수 있다.]

이리하여 $P(365,\ k)\geq 0.5$일 최소의 k값을 보자. 우선 $Q(365,\ k)$로 임명한 중복 없는 확률을 유도하는것은 아주 쉽다. $k>365$이면 모든 값이 서로 다르다는것은 불가능하다. 그래서 $k\leq 365$라고 가정한다. 이제는 중복 없는 k개 값을 가질수 있는 서로 다른 방법의 수 N을 고찰하자. 첫째로 365개 값중 임의의 하나를 택하고 둘째로 나머지 364개 수중의 임의의 하나를 택하며 이런 과정을 반복한다. 따라서 서로 다른 방법의 수는

$$N = 365 \times 364 \times \cdots \times (365 - k + 1) = \frac{365!}{(365-k)!} \qquad (8\text{-}2)$$

이다.

중복이 없다는 제한을 제거하면 매번 365개 값중의 하나를 택하며 총 확률의 수는 365^k이다. 그래서 중복 없는 확률은 단순히 다음과 같은 값모임모두의 밖에서 중복이 없는 값모임들의 분수이다.

$$Q(365, k) = \frac{365!/(365-k)!}{(365)^k} = \frac{365!}{(365-k)!(365)^k} \text{ 이고}$$

$$P(365, k) = 1 - Q(365-k) = 1 - \frac{365!}{(365-k)!(365)^k} \qquad (8\text{-}3)$$

이 함수를 그림 8-11에서 보여 준다. 이미전에 이 문제를 고찰하지 않은 사람들에게는 이 확률이 놀랍게도 크게 보일것이다. 대부분의 사람들은 적어도 하나의 중복이 있는 0.5이상의 확률을 가지기 위해서는 그 집단의 사람수가 대략 100이여야 한다고 추측할것이다. 사실 그 수는 23이며 P(365, 23)＝0.5073이다. k＝100에 대하여 적어도 하나의 중복이 있을 확률은 0.9999997이다.

아마도 집단안의 특수한 사람을 고찰할 때 결과가 그렇게 놀랍게 보이는 원인은 그 집단안의 어떤 다른 사람이 같은 생일을 가질 확률이 작은것이다. 그러나 관련된 확률은 집단안의 임의의 사람쌍이 같은 생일을 가질 확률이다. 23명의 집단에서는 $\frac{23(23-1)}{2}$＝253개의 서로 다른 사람쌍이 있다. 따라서 확률이 높다.

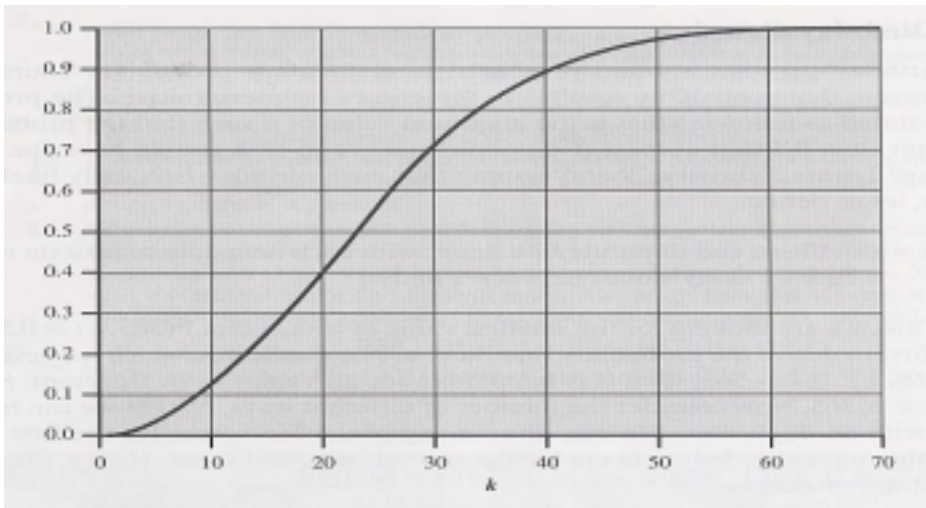



그림 8-11. 생일모순

유용한 부등식

생일문제의 일반화를 개발하기전에 다음과 같은 필요한 부등식을 유도한다. 모든 $x \geq 0$에 대하여

$$(1-x) \leq e^{-x} \tag{8-4}$$

그림 8-12는 이 부등식을 보여 준다. 부등식이 성립함을 보기 위하여 아래선이 $x=0$에서 e^{-x}에 대한 탕겐스임을 강조한다. 그 선의 경사도는 바로 $x=0$에서 e^{-x}의 도함수이다. 즉

$$f(x)=e^{-x}, \qquad f'(x)=\frac{d}{dx}e^{-x}=e^{-x}, \qquad f'(0)=-1$$

탕겐스는 $ax+b$형태의 직선이며 $a=-1$과 $x=0$에서의 탕겐스는 $e^{-0}=1$이여야 한다. 이리하여 부등식 8-4를 확신하는 함수 $(1-x)$가 탕겐스이다. 더 나아가서 작은 x에 대하여 $(1-x) \approx e^{-x}$이라는것을 강조한다.

중복이 있는 일반적경우

생일문제를 다음과 같은 문제로 일반화할수 있다. 1과 n사이의 평등분포하는 우연변수와 우연변수의 k개 실체$(k \leq n)$의 선택이 주어 졌을 때 적어도 하나의 중복이 있을 확률 $P(n,\ k)$는 무엇인가? 생일문제는 바로 $n=365$인 특수경우이다. 앞에서와 같은 리유로 하여 식 8-3의 다음과 같은 일반화를 얻는다.

$$P(n,\ k)=1-\frac{n!}{(n-k)!n^k} \tag{8-5}$$

다음과 같이 쓸수 있다.

$$\begin{aligned} P(n,k) &= 1-\frac{n\times(n-1)\times\ldots\times(n-k+1)}{n^k} \\ &= 1-\left[\frac{n-1}{n}\times\frac{n-2}{n}\times\ldots\times\frac{n-k+1}{n}\right] \\ &= 1-\left[(1-\frac{1}{n})\times(1-\frac{2}{n})\times\ldots\times(1-\frac{k-1}{n})\right] \end{aligned}$$

부등식 8-4를 리용하면

$$\begin{aligned} P(n,k) &> 1-[(e^{-1/n})\times(e^{-2/n})\times\ \cdots\times(e^{-(k-1)/n})] \\ &> 1-e^{-[(1/n)+(2/n)+\cdots+((k-1)/n)]} \\ &> 1-e^{-k\times(k-1)/2n} \end{aligned}$$

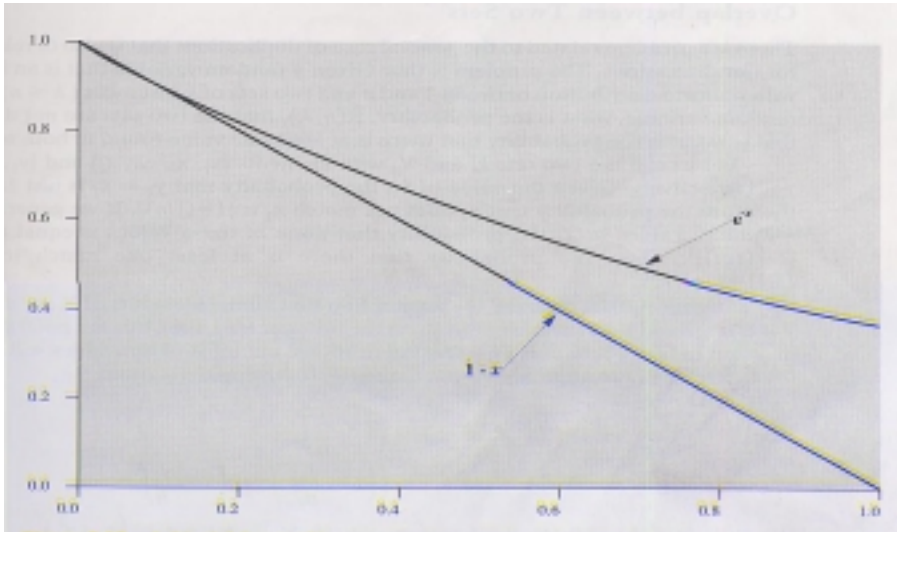



그림 8-12. 유용한 부등식

질문을 하나 제기하자. k의 값이 $P(n, k)>0.5$이도록 요구하는것은 무엇인가? 이 요구를 만족시키기 위하여서는

$$1/2=1-e^{-(k\times(k-1))/2n}$$
$$2=e^{(k\times(k-1))/2n}$$
$$\ln(2)=\frac{k\times(k-1)}{2n}$$

큰 k에 대하여 $k\times(k-1)$를 k^2으로 바꾸어 놓으면

$$k=\sqrt{2(\ln 2)n}=1.18\sqrt{n}\approx\sqrt{n} \qquad (8\text{-}6)$$

이것을 확인하기 위하여 $n=365$일 때 $k=1.18\times\sqrt{365}=22.54$로서 23이라는 정확한 대답에 아주 가깝다. 이제는 다음과 같은 용어로 생일공격의 기초를 규정할수 있다. 2^m개의 가능한 출력(즉 $m$bit출구)을 가지는 함수 H가 있다고 가정하자. H를 k개의 우연입력에 적용하면 적어도 하나의 중복(즉 어떤 입력 x, y에 대하여 $H(x)=H(y)$)을 가질 확률이 있을 k의 값은 무엇이여야 하는가? 식 8-6의 근사식을 리용하면

$$k=\sqrt{2^m}=2^{m/2} \qquad (8\text{-}7)$$

두 모임사이의 겹침

우리의 론의에 관련되는것으로서 중복이 있는 일반적인 경우에 관계되는 문제가 하

나 있다. 이것은 다음과 같은 문제이다. 1과 n사이의 평등분포하는 우연옹근변수와 이 우연변수의 k개의 실체($k \le n$)들의 두 모임이 주어 졌을 때 두 모임이 비분리적일 확률 R(n, k)는 무엇인가 즉 두 모임에 속하는 적어도 하나의 값이 있을 확률은 무엇인가?

두 모임 X와 Y를 각각 $\{x_1, \cdots, x_k\}$와 $\{y_1, \cdots, y_k\}$이라고 하자. x_1의 값이 주어 졌을 때 $x_1 = y_1$일 확률은 바로 $1/n$이므로 y_1가 x_1를 정합하지 못할 확률은 $[1-(1/n)]$이다. Y의 k개우연값을 생성하면 이 값들의 어느것이나 x_1와 같을 확률은 $[1-(1/n)]^k$이다. 이리하여 x_1를 정합하는것이 적어도 하나 있을 확률은 $1-[1-(1/n)]^k$이다.

계속하여 X의 모든 원소들이 서로 다르다고 가정하자. n이 크고 k역시 크면 (즉 $\sqrt{n}$정도이라면) 이것은 좋은 근사이다. 사실 아주 적은 중복이 있으나 대부분의 값들은 서로 다를것이다. 이 가정밑에 다음과 같은것을 유도할수 있다.

$$\Pr[\text{Y에서는 } x_1\text{를 정합못함}] = \left(1-\frac{1}{n}\right)^k$$

$$\Pr[\text{Y에서는 X를 정합못함}] = \left(\left(1-\frac{1}{n}\right)^k\right)^k = \left(1-\frac{1}{n}\right)^{k^2}$$

$$R(n,k) = \Pr[\text{Y에서는 X를 정합할 적어도 하나가 있음}] = 1-\left(1-\frac{1}{n}\right)^{k^2}$$

부등식 8-4를 리용하면

$$R(n,k) > 1-\left(e^{-1/n}\right)^{k^2}$$
$$R(n,k) > 1-\left(e^{-k^2/n}\right)$$

질문을 하나 제기하자. $R(n,k)>0.5$일것을 요구하는 k의 값은 무엇인가? 이 요구를 만족시키기 위하여

$$1/2 = 1-\left(e^{-1/n}\right)^{k^2}$$
$$2 = e^{k^2/n} \qquad (8\text{-}8)$$
$$\ln(2) = \frac{k^2}{n}$$
$$k = \sqrt{(\ln(2))n} = 0.83\sqrt{n} \approx \sqrt{n}$$

이것을 다음과 같이 생일공격에 관계되는 용어로 규정할수 있다. 2^m개의 가능한 출력(즉 $m$bit출력)을 가지는 함수 H가 있다고 가정하자. H를 k개 우연입력에 적용하여 모임 X를 생성해 내고 다시 k개의 보충적인 우연입력에 적용하여 모임 Y를 생성해 낸다. 두 모임사이의 정합을 이루는것이 적어도 하나 있을(즉 어떤 입력 $x \in X$, $y \in Y$에 대하여 $H(x)=H(y)$)확률이 있을 k의 값은 무엇이여야 하는가? 근사식 8-8을 리용하면

$$k = \sqrt{2^m} = 2^{m/2}$$

제9장. 하쉬 및 Mac알고리듬

하쉬함수의 전개와 대칭블로크암호의 전개에는 일련의 류사성이 있다. 힘내기공격의 증가능력과 암호분석의 발전은 DES의 인기를 떨어 뜨렸으며 특수한 암호분석공격을 방지하기 위하여 많은 기능과 보다 긴 열쇠길이를 가진 새로운 알고리듬을 설계하게 하였다. 마찬가지로 계산능력과 하쉬함수암호분석의 발전은 두개의 아주 극적인 하쉬함수 MD4와 MD5의 인기를 떨어 뜨렸다. 이에 대한 도전으로서 보다 새로운 하쉬알고리듬은 보다 긴 하쉬부호로 발전되였으며 특수한 암호분석공격을 방지하기 위하여 많은 기능을 가지도록 설계되였다. 다른 하나의 류사성은 증명된 구조로부터 떨어 지지 않으려고 하는것이다. DES는 페이스텔(Feistel)암호에 기초하고 있는데 사실 이 암호는 샤농(Shannon)의 치환망제의에 기초하고 있다. 실제적으로 중요하고 본질적인 블로크암호는 페이스텔설계를 따른다. 왜냐하면 그 설계가 새롭게 발견된 암호분석위협을 방지하는데 적합하기때문이다. 그 대신에 완전히 새로운 설계가 대칭블로크암호에 리용되였다면 구조자체가 아직은 생각하지 못한 새로운 공격의 길을 열어 놓았을것이라는 생각도 있다. 마찬가지로 대부분의 중요한 현대하쉬함수들은 그림 8-10의 기본구조를 따른다. 다시 이것은 기본적으로 건전한 구조이라는것을 증명하였으며 보다 새로운 설계는 단순히 그 구조를 세분화하며 하쉬부호길이에 첨부하였다.

이 장에서는 3개의 중요한 하쉬함수 MD5, SHA-1과 RIPEMD-160을 본다. 그다음에 하쉬함수의 리용에 기초하는 인터네트표준통보문인증부호 HMAC를 본다.

9.1 MD5통보문요약정보알고리듬

MD5통보문요약알고리듬(RFC 1321)은 론 리베스트(Ron Rivest)가 MIT에서 개발하였다. 최근 몇년전까지 폭력과 암호분석이 둘 다 제기될 때 MD5가 가장 안전하게 리용된 안전한 하쉬알고리듬이다.

MD5 론리

이 알고리듬은 임의의 길이의 통보문을 입력으로 취하여 128bit통보문요약을 출력으로 내보낸다. 입력은 512bit블로크내에서 처리된다.

그림 9-1은 요약을 생성해 내는 통보문의 전반적처리를 보여 준다. 이것은 그림 8-10에서 보여 준 일반구조를 따른다. 처리는 다음의 단계들로 이루어 진다.

- **단계1: 메꾸기비트의 첨가.** 통보문을 그 비트길이가 mod 512에 관하여 448과 합동(길이≡448 mod 512)이도록 공백으로 메꾼다. 즉 메꾸어 진 통보문이 이미 요구된 길이비트의 옹근수배수보다 적은 64bit이다. 통보문이 이미 요구된 길이일 때도 메꾸기를 항상 첨가한다. 실례로 통보문이 448bit이면 512bit를 메꾸기하여 960bit의 길이로 된다. 이리하여 메꾸어 지는 비트수는 1부터 512사이에 놓인다.

  메꾸기는 필요한 수의 0-bit에 의해 뒤따르는 유일한 1-bit로 이루어 진다.

- **단계2: 길이의 첨가.** (메꾸기전에) 원래 통보문의 비트길이의 64bit표현은 단계의 결과(최소의 유효한 바이트를 우선)에 첨가된다. 원래의 길이가 2^{64}보다 크면 오직 길이의 낮은자리 64bit만을 리용한다. 이리하여 그 마당은 mod 2^{64}에 관하여 원래통보문의 길이를 포함한다. 첫 두단계의 결과는 512bit길이의 옹근수인 통보문을 산출한다. 그림 9-1에서는 확장된 통보문을 512bit블로크렬 Y_0, Y_1, …, Y_{L-1}로 표현하여 확장된 통보문의 총 길이는 $L\times512$bit이다. 동등하게 결과는 1632bit단어의 배수이다. M[0 … N-1]은 N이 16의 옹근수배수인 결과통보문의 단어를 표시한다고 하자. 이리하여 $N=L\times16$이다.

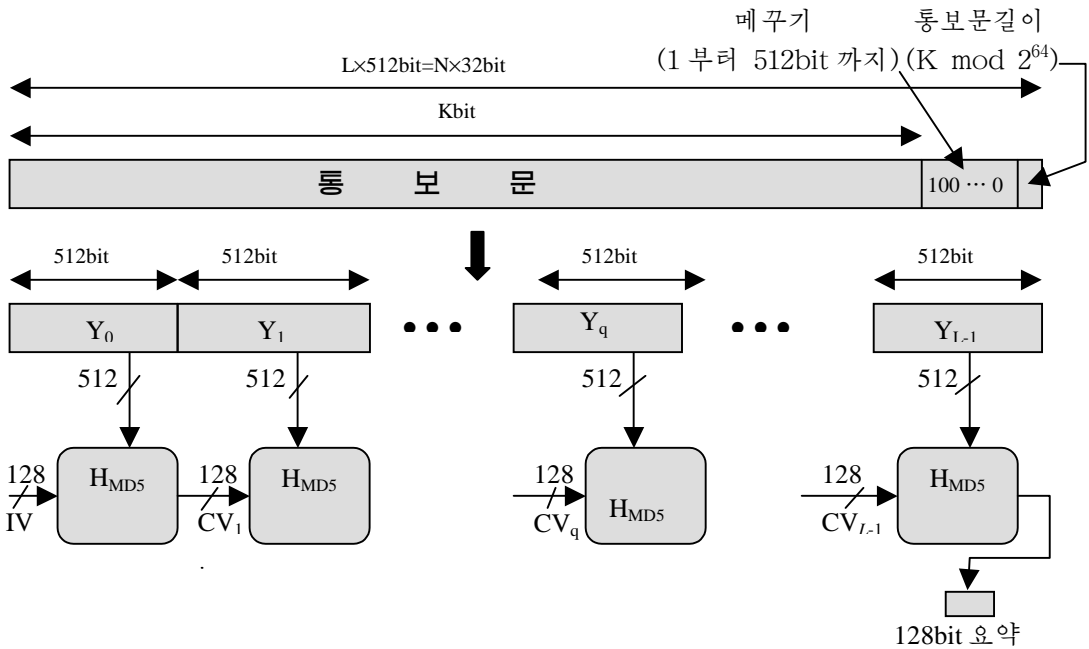



그림 9-1. MD5를 리용한 통보문요약생성

- **단계3: MD완충기의 초기화.** 128bit완충기는 하쉬함수의 중간 및 최종결과를 보관하는데 리용된다. 완충기를 4개의 32bit등록기(A, B, C, D)로 표현할수 있다. 이 등록기를 다음과 같은 32bit옹근수(16진)로 초기화한다.

A=67452301
B=EFCDAB89
C=98BADCFE
D=10325476

이 값들을 작은 endian양식속에 기억시키는데 이것은 낮은 주소바이트위치에서 단어의 최소유효바이트이다. 32bit렬의 초기화값(16진)은 다음과 같다.

word A: 01 23 45 67
word B: 89 AB CD EF
word C: FE DC BA 98
word D: 76 54 32 10

- **단계4: 512bit(16-단어)블로크의 처리통보문.** 알고리듬의 핵심부는 다음과 같은 4개의 처리《회전》으로 이루어 진 압축함수이다. 이 모듈을 그림 9-1에서 H_{MD5}으로 표시하며 그의 론리를 그림 9-2에서 보여 준다. 4개의 회전은 류사한 구조를 가지지만 매개는 서로 다른 원시국부함수를 리용하는데 이것들은 그림에서 F, G, H 그리고 I라고 하였다.

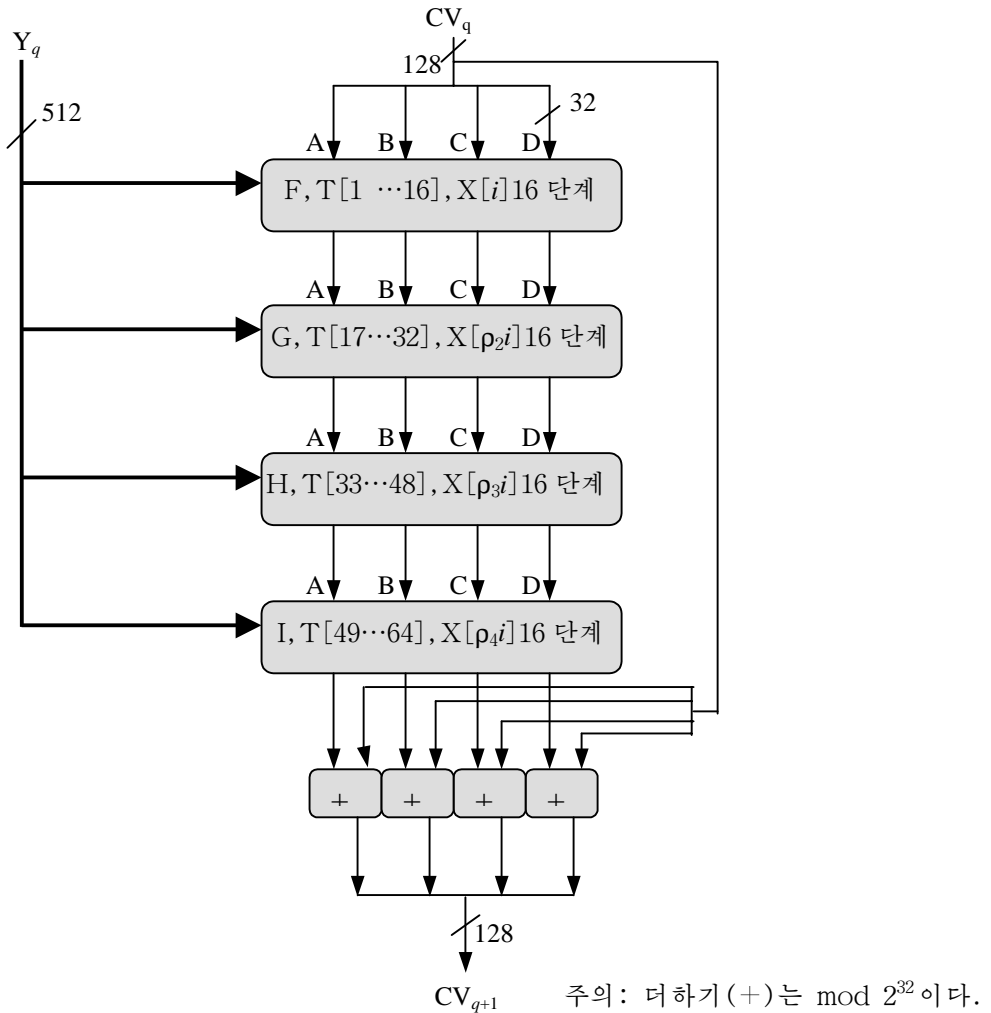



그림 9-2. 하나의 512bit의 블로크의 MD5처리(MD5압축함수)

매 회전은 (Y_q)를 처리하는 현재의 512bit블로크와 128bit완충기값 ABCD를 입력으로 취하여 완충기의 내용을 갱신한다. 매 회전은 또한 시누스함수로 구성된 64개 요소표 T[1 ... 64]의 1/4을 리용한다. T의 i번째 요소 T[i]는 $2^{32}\times$abs(sin(i))의 옹근수부이다. 여기서 i는 라디안이다. abs(sin(i))는 0과 1사이 수이므로 T의 매 요소는 32bit로 표현할수 있는 옹근수이다. 표는 32bit패턴들의 《우연화된》모임을 제공하는 바 입력자료에서 임의의 정칙성을 소거한다. 표 9-1의 ㄴ은 T의 값을 렬거한다.

표 9-1. MD5의 열쇠요소

B	c	D	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

ㄱ) 론리함수의 진리표

T[1]=D76AA478	T[17]=F61E2562	T[33]=FFFA3942	T[49]=F4292244
T[2]=E8C7B756	T[18]=C040B340	T[34]=8771F681	T[50]=432AFF97
T[3]=242070DB	T[19]=256E5A51	T[35]=699D6122	T[51]=AB9423A7
T[4]=C1BDCEEE	T[20]=E9B6C7AA	T[36]=FDE5380C	T[52]=FC93A039
T[5]=F57COFAF	T[21]=D62F105D	T[37]=A4BEEA44	T[53]=655B59C3
T[6]=4787C62A	T[22]=02441453	T[38]=4BDECFA9	T[54]=8F0CCC92
T[7]=A8304613	T[23]=D8A1E681	T[39]=F6BB4B60	T[55]=FFEFF47D
T[8]=FD469501	T[24]=E7D3FBC8	T[40]=BEBFBC70	T[56]=85845DD1
T[9]=698098D8	T[25]=21E1CDE6	T[41]=E89B7EC6	T[57]=6FA87E4F
T[10]=8B44F7AF	T[26]=C33707D6	T[42]=EAA127FA	T[58]=FE2CE6E0
T[11]=FFFF5BB1	T[27]=F4D50D87	T[43]=D4EF3085	T[59]=A3014314
T[12]=895CD7BE	T[28]=455A14ED	T[44]=04881D05	T[60]=4E0811A1
T[13]=6B901122	T[29]=A9E3E905	T[45]=D9D4D039	T[61]=F7537E82
T[14]=FD987193	T[30]=FCEFA3F8	T[46]=E6DB99E5	T[62]=BD3AF235
T[15]=A679438E	T[31]=676F02D9	T[47]=1FA27CF8	T[63]=2AD7D2BB
T[16]=49B40821	T[32]=8D2A4C8A	T[48]=C4AC5665	T[64]=EB86D391

ㄴ) 시누스함수들로 구성된 표 T

네번째 회전의 출력은 CV_{q+1}를 생성해 내는 첫 회전(CV_q)의 입력에 첨가된다. 더하기는 mod 2^{32}에 관한 더하기를 리용하여 CV_q의 대응하는 매 단어와 함께 완충기의 4개의 매 단어에 대하여 독립적으로 진행된다.

- **단계5: 출력.** 모든 L다음에 512bit블로크를 처리하며 L번째 단계의 출력은 128bit통보문요약이다.

MD5의 작용을 다음과 같이 개괄할수 있다.

$$CV_0 = IV$$
$$CV_{q+1} = SUM_{32}(CV_q, RF_I[Y_q, RF_H[Y_q, RF_G[Y_q, RF_F[Y_q, CV_q]]]])$$
$$MD = CV_L$$

여기서

IV : 단계3에서 정의된 완충기의 초기값
Y_q : 통보문의 q번째 512bit블로크
L : 통보문에서 블로크수(메꾸기와 길이마당포함)
CV_q : 통보문의 q번째 블로크와 함께 처리된 련쇄변수
RF_x : 원시국부함수 x를 리용한 회전함수
MD : 최종통보문요약값
SUM_{32}: 입력쌍의 매 단어우에서 개별적으로 진행된 mod 2^{32}에 관한 더하기

MD5 압축함수

하나의 512bit블로크처리의 4개의 매 회전에 대한 론리를 보다 구체적으로 보자. 매 회전은 완충기 ABCD우에서 조작하는 16개 단계들의 렬로 이루어 진다. 매 단계의 형태는

$$a \leftarrow b + ((a + g(b, c, d) + X[k] + T[i]) <<< s$$

여기서

a, b, c, d: 단계에 따라 변하는 순서로 서술된 완충기의 4개 단어
g : 원시함수 F, G, H, I중의 하나.
<<<s : s개 비트에 의한 32bit인자의 왼쪽순환밀기(회전)
$X[k]$: $M[q \times 16 + k]$:통보문의 q번째 512bit블로크에서 k번째 32bit단어
$T[i]$: 행렬 T에서 i번째 비트단어
$\times$: mod 2^{32}에 관한 더하기

그림 9-3은 단계연산을 보여 준다. 4개의 단어 (a, b, c, d)를 리용하는 순서는 매 단계에 대하여 한 단어의 단어수준오른쪽순환밀기를 생성해 낸다.

4개의 원시국부함수들중의 하나는 알고리듬의 4개의 매 회전에 리용된다 . 매 원시함수들은 3개의 32-bit단어를 입력으로 취하여 32-bit단어를 출력으로 내보낸다. 매 함수는 비트별 론리연산들의 모임을 진행한다. 즉 출력의 n번째 비트는 3개 입력의 n번째 비트에 관한 함수이다. 함수들을 다음과 같이 개괄할수 있다.

회전	원시함수 g	g(b,c,d)
1	F(b,c,d)	$(b \wedge c) \vee (\neg b \wedge d)$
2	G(b,c,d)	$(b \wedge d) \vee (c \wedge \neg d)$
3	H(b,c,d)	$b \oplus c \oplus d$
4	I(b,c,d)	$c \oplus (b \vee \neg d)$

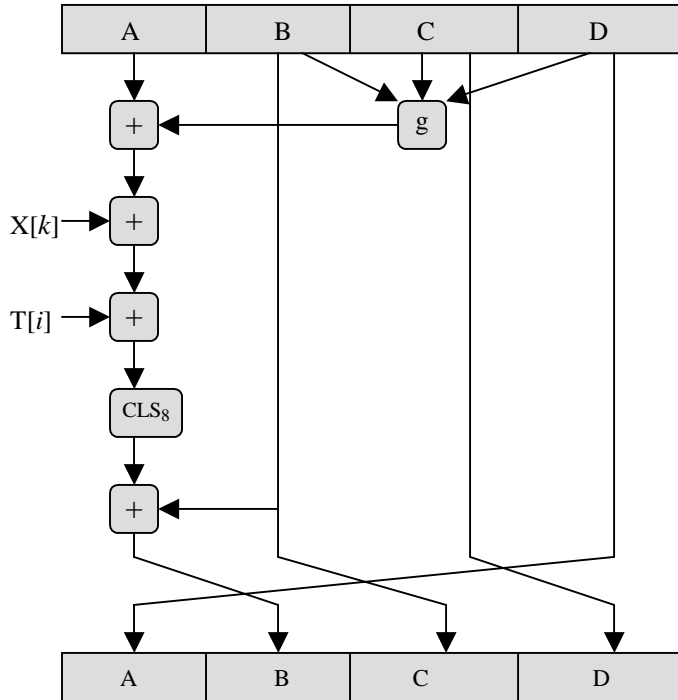



그림 9-3. 초등 MD5연산(한단계)

론리연산(AND,OR,NOT,XOR)들을 기호($\wedge, \vee, \neg, \oplus$)로 표시한다. 함수 F는 조건부함수이다. 즉 b이면 c아니면 d이다. 류사하게 G를 다음과 같이 정식화할수 있다. d이면 b아니면 c이다. 함수 H는 기우성비트를 생성해 낸다. 표 9-1의 ㄱ은 4개 함수의 진리표이다.

RFC 1321로부터 채용된 그림 9-4는 단계 4의 처리알고리듬을 정의한다. 32-bit단어 X[0…15]는 처리되는 현재의 512-bit입력블로크의 값을 취한다. 회전내에서 X[i]의 16개의 매 단어는 한단계동안 정확히 한번 리용된다. 그리고 이 단어들의 리용순서는 회전에 따라 변한다. 첫 회전에서 단어는 원래의 순서로 리용된다. 2회전부터 4회전까지는 다음의 치환을 정의한다.

$$\rho_2(i) = (1+5i) \bmod 16$$
$$\rho_3(i) = (5+3i) \bmod 16$$
$$\rho_4(i) = 7i \bmod 16$$

```
/*매 16-단어(512-bit)블로크처리*/
For q=0 to (N/16)-1 do
    /* 블로크 q를 X에 복사. */
    For j=0 to 15 do
          Set X[j] to M[q*16+j].
    end /* j에 관한 순환 */

    /* A,B,C,D를 각각
    AA,BB,CC,DD로 보관 */
    AA=A
    BB=B
    CC=C
    DD=D

    /* 1회전 */
    /* 연산
    a=b+((a+F(b,c,d)+X[k]+T[i])<<<
    s)을 [abcd k s i]로 표시한다. 다
    음의 16개 연산을 한다. */
    [ABCD    0    7    1]
    [DABC    1    12   2]
    [CDAB    2    17   3]
    [BCDA    3    22   4]
    [ABCD    4    7    5]
    [DABC    5    12   6]
    [CDAB    6    17   7]
    [BCDA    7    22   8]
    [ABCD    8    7    9]
    [DABC    9    12   10]
    [CDAB    10   17   11]
    [BCDA    11   22   12]
    [ABCD    12   7    13]
    [DABC    13   12   14]
    [CDAB    14   17   15]
    [BCDA    15   22   16]

    /* 2회전 */
    /* 연산
    a=b+((a+G(b,c,d)+X[k]+T[i])<<
    <s)을 [abcd k s i]로 표시한다.
    다음의 16개 연산을 한다. */
    [ABCD    1    5    17]
    [DABC    6    9    18]
    [CDAB    11   14   19]
    [BCDA    0    20   20]
    [ABCD    5    5    21]
    [DABC    10   9    22]
    [CDAB    15   14   23]
    [BCDA    4    20   24]
    [ABCD    9    5    25]
    [DABC    14   9    26]
    [CDAB    3    14   27]
    [BCDA    8    14   27]
    [ABCD    13   5    29]
    [DABC    2    9    30]
    [CDAB    7    14   31]
    [BCDA    12   20   32]

    /* 3회전 */
    /* 연산
    a=b+((a+H(b,c,d)+X[k]+T[i])<<
    <s)을 [abcd k s i]로 표시한다.
    다음의 16개 연산을 한다. */
    [ABCD    5    4    33]
    [DABC    8    11   34]
    [CDAB    11   16   35]
    [BCDA    14   23   36]
    [ABCD    1    4    37]
    [DABC    4    11   38]
    [CDAB    7    16   39]
    [BCDA    10   23   40]
    [ABCD    13   4    41]
    [DABC    0    11   42]
    [CDAB    3    16   43]
    [BCDA    6    23   44]
    [ABCD    9    4    45]
    [DABC    12   11   46]
    [CDAB    15   16   47]
    [BCDA    2    23   48]

    /* 4회전 */
    /* 연산
    a=b+((a+I(b,c,d)+X[k]+T[i])<<<
    s)을 [abcd k s i]로 표시한다. 다
    음의 16개 연산을 한다. */
    [ABCD    0    6    49]
    [DABC    7    10   50]
    [CDAB    14   15   51]
    [BCDA    5    21   52]
    [ABCD    12   6    53]
```

```
[DABC   3    10   54]          /* 4개 등록기 매개가 이 블로크를
[CDAB   10   15   55]          시작하기전에 가진 값들로 매 등록
[BCDA   1    21   56]          기의 증가 */
[ABCD   8    6    57]          A=A+AA
[DABC   15   10   58]          B=B+BB
[BCDA   13   21   60]          C=C+CC
[ABCD   4    6    61]          D=D+DD
[DABC   11   10   62]      end /* q에 관한 순환 */
[CDAB   2    15   63]
[BCDA   9    21   64]
```

그림 9-4. 기본 MD5갱신알고리듬(RFC 1321)

T의 64개의 32bit단어요소 매개는 한회전 한단계동안 정확히 한번 리용된다. 또한 매 단계에 대하여 오직 ABCD완충기의 4개 바이트중 하나만이 갱신된다는것을 강조한다. 따라서 완충기의 매 바이트는 회전당 4번 갱신되며 그다음 5번째에는 이 블로크에 대한 최종값을 생성해 내는 마감에 갱신된다. 최종적으로 4개의 각이한 왼쪽순환밀기량은 매 회전에서 리용되며 회전에 따라 다르다. 이 복잡성모두는 충돌(같은 출력을 생성해 내는 2개의 512bit블로크)을 생성하는것이 아주 힘들다는것이다.

MD4

MD4는 같은 설계가 론 리베스트가 개발한 MD5의 전신이다. 이것은 원래 1990년 10월에 RFC로 발표되였다. 약간 개정된 판이 1992년 4월에 MD5와 같은 날자에 RFC 1320으로 발표되였다. MD5는 리베스트의 문헌[RIVE90]을 문서화한 MD4의 설계목표를 공유하므로 간단히 MD4를 론의한다. 다음과 같은 목표를 렬거하였다.

- **보안:** 하쉬부호에 대한 유용한 요구가 있다. 다시 말하여 같은 통보문요약을 가지는 두 통보문을 구하는것은 계산량적으로 불가능하다는것이다. 이것의 발표시각에 요약의 길이때문에 MD4는 힘내기공격에 안전하였다. 리베스트는 또한 기술상태와 MD4의 복잡상태에 기초한 암호분석공격에 안전하다는것을 느끼였다.
- **속도:** 알고리듬은 고속으로 실행하는 쏘프트웨어에 의한 실현에 그자체를 제공한다. 특히 알고리듬은 32bit방식에서 고속일것이다. 이리하여 알고리듬은 32bit단어우에서 원시연산들의 간단한 모임에 기초하고 있다.
- **단순성과 조밀성:** 큰 프로그람을 요구하거나 치환표를 요구함이 없이 알고리듬을 서술하고 관리하는것이 단순하다. 이런 특성들은 뚜렷한 프로그람작성우점을 가질뿐아니라 보안의 관점으로부터 적합하다. 왜냐하면 단순한 알고리듬은 필요한 판정기준을 얻는데 더 적합하기때문이다.
- **favor-little-endian방식:** 일부 처리방식(Intel 80xxx와 Pentium계렬과 같은것)들은 단어의 최소유효바이트를 낮은 주소바이트위치(little-endian)에 기억한다. 다른 방식(SUN Sparcstation)들은 단어의 가장 유효한 바이트를 낮은 주소바이트위치(big-endian)에 기억한다. 이 차이는 통보문을 32bit단어들의 렬을 취급할때 유효하다. 왜냐하면 두 방식중의 하나는 처리를 위해 매 단어에서 바이트를 반

전해야 하기때문이다. 리베스트는 통보문을 32bit단어들의 렬로 해석하기 위해 little-endian방식을 선택하였다. 이 선택은 big-endian처리가 일반적으로 고속이라는 리베스트의 관찰에 기초하고 있으므로 처리법칙을 더 좋게 줄수 있다.

이 설계목표는 MD5에서 수행되였다. MD5는 좀 더 복잡한것이므로 MD4보다 실행속도가 좀 뜨다. 리베스트는 첨가된 복잡성이 주어 진 보안의 증가수준에 의해 정당화되였다는것을 느끼였다. 다음과 같은것이 둘사이의 주요차이이다.

1. MD4는 매개가 16개 단계들로 이루어 진 3개 회전을 리용한다면 MD5는 매개가 16개 단계들로 이루어진 4개 회전을 리용한다.
2. MD4에서는 첫 회전에서 보충적인 상수를 리용하지 않는다. 세번째 회전의 매 단계에서 다른 하나의 보충적인 상수를 리용한다. MD5에서는 서로 다른 보충적인 상수 T[i]를 64개의 매 단계에서 리용한다.
3. MD5는 4개의 원시론리함수를 리용하여 매 회전에서 이것들은 MD4에서의 3개와 비교되여 다시 매 회전에서 이것들을 리용한다.
4. MD5에서는 매 단계가 선행단계의 결과에 보충된다. 실례로 단계 1의 결과는 단어 A를 갱신한다. D에 기억된 단계 2의 결과는 A를 왼쪽순환밀기로 보충함으로써 형성된다. 마찬가지로 단계 3의 결과는 C에 기억되며 D를 왼쪽순환밀기결과에 보충함으로써 형성된다. MD4는 이 최종보충을 포함하지 않는다. 리베스트는 선행단계의 결과의 포함이 보다 큰 사태의 효과를 조장시킨다는것을 느끼였다.

MD5의 강도

MD5알고리듬은 하쉬부호의 매 비트가 입력안에서의 매 비트에 관한 함수이라는 성질을 가진다. 기초함수(F,G,H,I)의 복잡한 반복은 잘 혼합된 결과를 내보낸다. 즉 우연적으로 선택된 두 통보문이 류사한 규칙성을 가진다고 할지라도 같은 하쉬부호를 가질것이라는데는 적합하지 않다. 리베스트는 128bit하쉬부호에 대하여 가능한껏 MD5가 강하다는것을 RFC로 추측한다. 다시말하여 같은 통보문요약을 가지는 두 통보문을 받을 어려움성은 2^{64}번의 연산의 준위이라면 주어 진 요약을 가진 통보문을 구할 어려움성은 2^{128}번의 연산의 준위이다.

이 작성에서 그 어떤 분석도 이런 추측을 발견하지 못한다. 그러나 MD5에 대한 공격에는 다음과 같은 불길한 경향도 있었다.

1. 버손(Berson)의 문헌[BERS92]은 서로 다른 암호분석을 리용하여 하나의 회전 MD5에 대한 같은 요약을 생성해 내는 두 통보문을 구하는것이 적당한 시간내에 가능하다는것을 보여 주었다. 4개 회전중 매개에 대하여 결과를 보여 주었다. 그러나 저자는 완전한 4개 회전 MD5에 대한 공격을 어떻게 일반화하는가를 보여 줄수 없었다.
2. 보어(Boer)와 보슬래즈(Bosselaers)의 문헌[BOER93]은 통보문블로크 X와 같은 출력상태를 산생하는 2개의 관련된 련쇄변수를 어떻게 구하는가를 보여 주었다. 즉 512bit의 단일블로크에서 MD5의 실행은 완충기 ABCD안에 두개의 서로 다른 입력값에 대한 같은 출력을 산생할것이다. 이것을 준충돌(pseudocollision)이라고 부른다. 현재 MD5의 성공적인 공격에 대한 이 방식을 확장하는 방법은 없는것 같다.

3. MD5에 대한 가장 심각한 공격을 도버틴(Dobbertin)의 문헌[DOBB96a]이 개발하였다. 그의 기술은 MD5압축함수에 대한 충돌을 생성할수 있는것이다. 즉 이 공격은 같은 128bit출력을 생성해 내는 다른 블로크를 찾음으로써 하나의 512bit입력블로크우에서 MD5의 연산에 대하여 동작한다. 이런 작성에 대하여 MD5의 초기값(IV)을 리용하여 완전한 통보문에 대한 이 공격을 일반화하는것을 구하는 방법은 아직 없다. 그럼에도 불구하고 이 공격의 성과는 위안에 불과할뿐이다.

이리하여 암호분석의 관점으로부터 MD5는 이제는 약한것으로 고찰되여야 한다는것을 알게 되였다. 더 나아가서 힘내기공격의 관점으로부터 MD5는 이제는 2^{64}의 효과준위로 요구하는 생일공격에 약하다. 그러므로 보다 긴 하쉬부호를 가지며 암호분석의 알려진 방법들에 더 방어적인 하쉬함수와 극적인 MD5를 재배치할 필요가 있다. 두개의 후보 SHA-1과 RIPEMD-160이 나왔다. 이것들을 다음 두개 절에서 설명한다.

9.2 안전한 하쉬알고리듬

안전한 하쉬알고리듬(SHA)을 1993년에 민족표준 및 기술연구소(NIST)가 개발하였으며 련방정보처리표준(FIPS PUB 180)으로 발표하였다. 갱신판이 1995년에 FIPS PUB 180-1로 발행되였으며 일반적으로 SHA-1이라고 한다. SHA는 MD4알고리듬에 기초하고 있으며 그 설계는 MD4와 밀접히 관련되여 있다.

SHA-1 론리

이 알고리듬은 2^{64}bit보다 작은 최대길이를 가지는 통보문을 입력으로 취하여 160bit 통보문요약을 출력으로 내보낸다. 입력은 512bit블로크로 처리된다.

전면적인 통보문처리는 그림 9-1에서 보여 준 512bit의 블로크길이와 하쉬길이 그리고 160bit의 련쇄변수길이를 가지는 MD5의 구조를 따른다. 처리는 다음과 같은 관계들로 이루어 진다.

- **단계1: 메꾸기비트의 첨가.** 통보문의 길이가 mod 512에 관하여 448과 합동(길이≡448 mod 512)이도록 통보문을 메꾼다. 통보문이 이미 요구된 길이라고 할지라도 메꾸기는 항상 첨가된다. 이리하여 메꾸기비트의 수는 1부터 512사이다. 메꾸기는 0-bit의 필요한 수로 뒤따르는 하나의 1-bit로 이루어 진다.
- **단계2: 길이첨가.** 64bit블로그를 통보문에 첨가한다. 이 블로크를 부호 없는 64bit옹근수(가장 유효한 바이트의 첫번째)로 취급하며 이것은 (메꾸기를 하기전에)원래의 통보문의 길이를 포함한다.
- **단계3: MD완충기의 초기화.** 160bit완충기를 리용하여 하쉬함수의 중간결과와 최종결과를 취한다. 완충기를 5개의 32bit등록기(A, B, C, D, E)로 표현할수 있다. 이 등록기들을 다음과 같은 32bit옹근수(16진수)로 초기화한다.

A＝67452301
B＝EFCDAB89
C＝98BADCFE
D＝10325476
E＝C3D2E1F0

첫 4개 값들은 MD5에서 리용된것과 같다는것을 강조한다. 그러나 SHA-1인 경우에 이 값들은 big-endian양식에 기억되는데 이것은 낮은 주소바이트위치에서 단어의 가장 유효한 바이트이다. 32bit렬의 초기값은 다음과 같다.

word A: 67 45 23 01
word B: EF CD AB 89
word C: 98 BA DC FE
word D: 10 32 54 76
word E: C3 D2 E1 F0

- **단계4: 512bit(16단어)블로크로 통보문처리.** 알고리듬의 기본은 20개의 매 단계처리의 4개 회전들로 이루어 진 모듈이다. 그림 9-5에서 그 론리를 보여 준다. 4개의 회전은 류사한 구조를 가지지만 매개는 각각 f_1, f_2, f_3, f_4라고 한 서로 다른 원시론리함수를 리용한다.

매 회전은 (Y_q)를 처리하는 현재 512bit블로크와 160bit 완충기값 ABCD를 입력으로 취하여 완충기의 내용을 갱신한다. 매 회전은 또한 보충적인 상수 K_t를 리용한다. 여기서 $0 \le t \le 79$는 4개 회전에 대한 80개 단계들중의 하나이다. 사실 오직 4개의 서로 다른 상수들만을 리용한다. 그 값들은 16 및 10진수로 아래의 표와 같다.

단계수	16진수	옹근수부
$0 \le t \le 19$	K_t＝5A827999	$[2^{30} \times \sqrt{2}]$
$20 \le t \le 39$	K_t＝6ED9EBA1	$[2^{30} \times \sqrt{3}]$
$40 \le t \le 59$	K_t＝8F1BBCDC	$[2^{30} \times \sqrt{5}]$
$60 \le t \le 79$	K_t＝CA62C1D6	$[2^{30} \times \sqrt{10}]$

4번째 회전(18번째 단계)의 출력을 CV_{q+1}를 생성해 내는 첫 회전(CV_q)의 입력에 첨가한다. 첨가는 mod 2^{32}에 관한 더하기를 리용하여 CV_q에서 대응하는 매 단어를 가지는 완충기안의 5개의 매 단어에 대하여 독립적으로 한다.

- **단계5: 출력.** 모든 L개의 512bit블로크들을 처리한 다음에 L번째 계단으로부터의 출력은 160bit통보문요약이다.

SHA-1의 작용을 다음과 같이 개괄할수 있다.

$$CV_0 = IV$$
$$CV_{q+1} = SUM_{32}(CV_q, ABCDE_q)$$
$$MD = CV_L$$

여기서

IV　　　: 단계 3에서 정의된 ABCDE완충기의 초기값
$ABCDE_q$: q번째 통보문블로크의 처리의 마지막회전의 출력
L　　　: 메꾸기와 길이마당을 포함하는 통보문에서 블로크의 수
SUM_{32} : 입력쌍의 매 단어우에서 개별적으로 진행된 mod 2^{32}에 관한 더하기
MD　　　: 최종통보문요약값

SHA-1 압축함수

하나의 512bit블로크처리의 80개 회전의 매개에서 론리를 좀 더 구체적으로 보자. 매 회전은 다음과 같은 형태이다(그림 9-6).

$$A, B, C, D, E \leftarrow (E + f(t, B, C, D) + S^5(A) + W_t + K_t), A, S^{30}(B), C, D$$

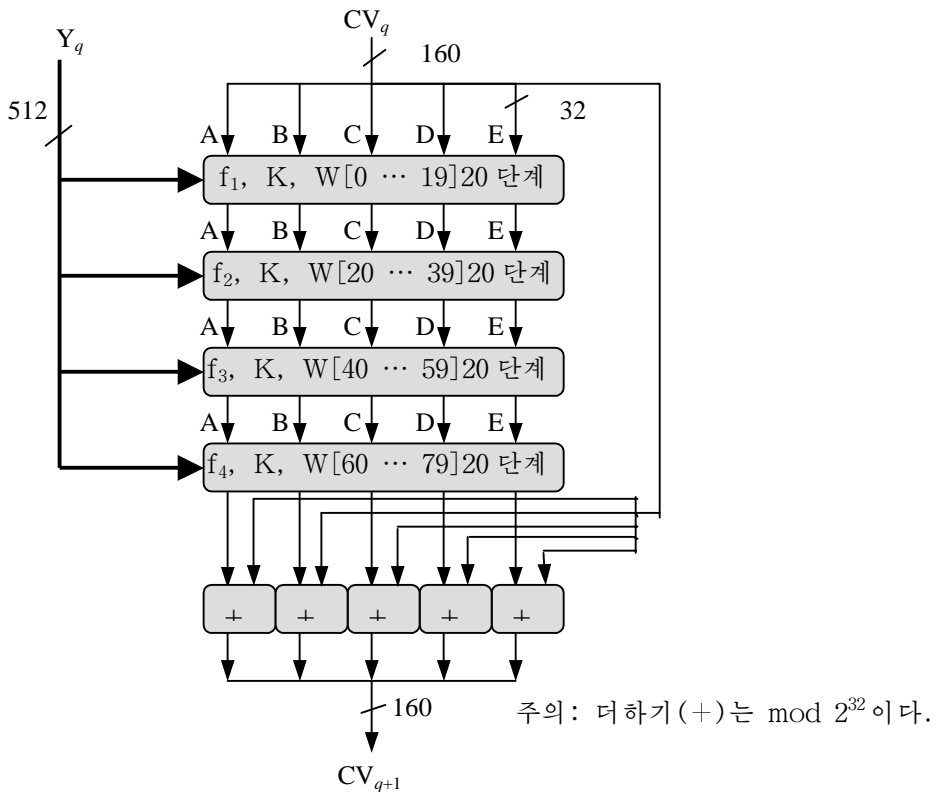



그림 9-5. 하나의 512bit블로크의 SHA-1처리(SHA-1압축함수)

여기서

A, B, C, D, E: 완충기의 5개 단어
t　　　: 단계수; $0 \le t \le 79$

$f(t, B, C, D)$: 단계 t에 대한 원시론리함수
S^k : k개 비트에 의한 32bit인자의 왼쪽순환밀기(회전)
W_t : 현재의 512bit입력블로크로부터 유도된 32bit단어
K_t : 더하기인자; 명백히 정의 된바와 같이 4개의 서로 다른 값들을 리용한다.
$+$: mod 2^{32}에 관한 더하기

매 원시함수는 3개의 32bit단어를 입력으로 취하여 32bit단어를 출력한다. 매 함수는 비트별 론리연산들의 모임을 진행한다. 즉 출력의 n번째 비트는 3개 입력의 n번째 비트에 관한 함수이다. 함수를 다음과 같이 개괄할수 있다.

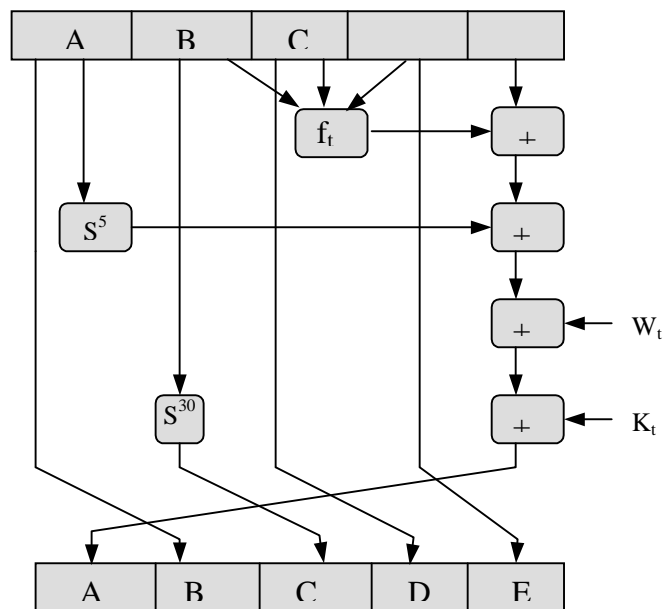



그림 9-6. 초등적인 SHA연산(하나의 단계)

단계수	함수이름	함수값
$(0\le t \le 19)$	$f_1=f(t, B, C, D)$	$(B\wedge C)\vee(B\wedge D)$
$(20\le t \le 39)$	$f_2=f(t, B, C, D)$	$B\oplus C\oplus D$
$(40\le t \le 59)$	$f_3=f(t, B, C, D)$	$(B\wedge C)\vee(B\wedge D)\vee(C\wedge D)$
$(60\le t \le 79)$	$f_4=f(t, B, C, D)$	$B\oplus C\oplus D$

론리연산자(AND, OR, NOT, XOR)들을 각각 기호(∧, ∨, ¬, ⊕)들로 표시한다. 볼수 있는바와 같이 오직 3개의 서로 다른 함수만을 리용한다. $0\le t \le 19$에 대하여 함수는 다음과 같은 조건부함수이다. 즉 B이면 C이고 아니면 D이다. $20\le t \le 39$와 $60\le t \le 79$에 대하여 함수는 기우성비트를 내보낸다. $40\le t \le 59$에 대하여 함수는 2 또는 3개의 인자들이 참이면 참이다. 표 9-2는 이 함수의 진리표이다.

32bit단어값 W_t를 512bit통보문으로부터 어떻게 유도하는가를 지적하는것이 남았다. 그림 9-7은 이 넘기기를 보여 준다. W_t의 첫 16개 값을 현재블로크의 16개 단어로부터 직접 취한다. 나머지 값들은 다음과 같이 정의한다.

$$W_t = S^1(W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3})$$

이리하여 처리의 첫 16개 단계에서 W_t의 값은 통보문블로크에서 대응하는 단어와 같다. 나머지 64개 단계에 대하여 W_t의 값은 W_t의 선행값중 4개의 XOR의 한 비트에 의해 왼쪽순환밀기로 이루어 진다. 이것은 MD5 및 RIPEMD-160과 뚜렷한 차이로서 이 두개는

표 9-2. SHA-1에 대한 론리함수의 자리표

B	C	D	$f_{0..19}$	$f_{20..39}$	$f_{40..59}$	$f_{60..79}$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	0	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	1

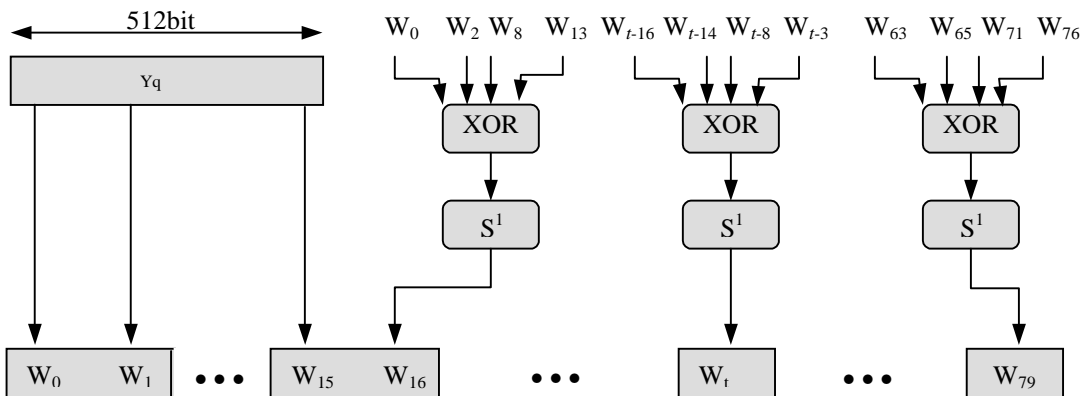



그림 9-7. 한 블로크의 SHA-1처리에 대한 80단어입력렬의 창조

매 단계함수에 입력으로서 직접 통보문블로크의 16개 단어중 하나를 리용한다. 그리고 오직 단어의 순서만을 회전에 따라 치환한다. SHA-1은 16개 블로크단어를 압축함수에서 리용하기 위하여 80개 단어로 확장한다. 이것은 통보문블로크에 과잉성과 호상관련성의 취급을 도입하게 하는데 통보문블로크들은 비교되여 같은 압축함수출력에로 넘기는 서로 다른 통보문블로크를 구하는 과제를 복잡하게 한다.

SHA-1과 MD5의 압축함수

둘 다 MD4로부터 유도되므로 SHA-1과 MD5는 다른 또 하나의것과 아주 류사하다. 따라서 그의 강도와 다른 특징들도 아주 류사하다. MD4에 대하여 앞에서 인용한 설계목표를 리용하여 두 알고리듬을 비교한다.

- **힘내기공격에 대한 보안:** 가장 명백하고 중요한 차이는 SHA-1요약이 MD5요약보다 32bit 더 길다는것이다. 폭력기술을 리용하여 주어 진 통보문요약을 가지는 임의의 통보문을 생성해 내는 어려움성은 MD5에 대해서는 2^{128}번의 연산준위이며 SHA-1에 대해서는 2^{160}번의 연산준위이다. 다시 폭력기술을 리용하여 같은 통보문요약을 가지는 두 통보문을 생성해 내는 어려움성은 MD5에 대해서는 2^{64}번의 연산준위이며 SHA-1에 대해서는 2^{80}번의 연산준위이다. 이리하여 SHA-1은 힘내기공격에 상당히 강하다.
- **암호분석에 대한 보안:** 앞의 절에서 론의한바와 같이 MD5는 그의 설계때부터 발견된 암호분석공격에 약하다. SHA-1은 이런 공격에 약하지 않다. 그러나 작은것이 SHA-1를 위한 설계판정기준에 대하여 공개적으로 알려 져서 그의 강도는 그렇지 않은 경우보다 평가하기가 더 힘들다.
- **속도:** 두 알고리듬이 심중하게 mod 2^{32}에 관한 더하기를 따르므로 둘 다 32bit방식으로 한다. SHA-1은 더 많은 단계(80 대 64)를 포함하며 MD5의 128bit완충기에 비교하여 160bit완충기를 처리해야 한다. 이리하여 SHA-1은 같은 하드웨어에 관하여 MD5보다 더 느리게 실행할것이다.
- **단순성과 콤팍트성:** 두 알고리듬은 서술하고 실현하는것이 단순하며 큰 프로그람이나 치환표를 요구하지 않는다.
- **little-endian대 big-endian방식:** MD5는 통보문을 32bit단어들의 렬을 해석하기 위하여 little-endian방식을 리용한다면 SHA-1은 big-endian방식을 리용한다. 두 방식보다도 더 발전된것은 아직 없는것 같다(이것이 적합하다. 왜냐하면 SHA의 NSA설계가들은 원형실현을 위해 Sun을 리용하였기때문이다).

9.3 RIPEMD-160

RIPEMD-160통보문요약알고리듬(문헌[DOBB96B, BOSS97])을 유럽 RACE완정성원시평가(RIPE)대상과제하에 MD4와 MD5에 대한 성과적인 공격을 부분적으로 시작한 연구집단이 개발하였다. 이 집단은 원래 128bit판의 RIPEM을 개발하였다. RIPE대상과제의 마감에 에취. 도버틴(RIPE대상과제의 한 성원이 아닌 사람)은 RIPEMD의 두 회전에 대한 공격을 발견하였으며 후에는 MD4와 MD5에 대한 공격을 발견하였다. 이 공격들때문에 RIPE협회의 일부 성원들로 RIPEMD를 갱신하기로 결정하였다. 설계작업을 그들과 도버틴이 하였다.

RIPEMD-160론리

이 알고리듬은 입력으로서 임의의 길이의 통보문을 취하여 출력으로서 160bit통보문

요약을 내보낸다. 입력은 512bit블로크로 처리된다.

통보문의 전면적인 처리는 그림 9-1에서 512bit의 블로크길이와 하쉬길이 그리고 160bit의 련쇄변수길이를 가지는 MD5에 대하여 보여 준 구조를 따른다. 처리는 다음과 같은 단계들로 이루어 진다.

- **단계1: 메꾸기비트의 첨가.** 통보문의 길이가 mod 512에 관하여 448과 합동(길이 ≡ 448 mod 512)이 되도록 통보문을 메꾼다. 통보문이 이미 요구하는 길이로 될지라도 메꾸기는 항상 첨가된다. 이리하여 메꾸기비트의 수는 1부터 512사이이다. 메꾸기는 0-bit의 필요한 수에 의해 뒤따르는 하나의 1-bit로 이루어 진다.
- **단계2: 길이첨가.** 64bit의 블로크를 통보문에 첨가한다. 이 블로크를 부호 없는 64bit옹근수(최소유효바이트의 첫번째)로 취급하며 이것은(메꾸기전에) 원래 통보문의 길이를 포함한다. MD5에서처럼 그리고 SHA-1과 대비하여 보면 RIPEMD-160은 little-endian협약을 리용한다.
- **단계3: MD완충기의 초기화.** 160bit완충기를 리용하여 하쉬함수의 중간결과와 최종결과를 얻는다. 완충기를 5개의 32bit등록기(A, B, C, D, E)로 표현할수 있다. 이런 등록기들은 다음과 같은 16진수값들로 초기화된다.

  A=67452301
  B=EFCDAB89
  C=98BADCFE
  D=10325476
  E=C3D2E1F0

  이것들은 SHA-1에서 리용된것들과 같은 값들이며 첫 4개는 MD5에서 리용된 것들과 같은 값이다. MD5에서와 같이 이 값들을 little-endian양식에 기억시킨다.

- **단계4: 512bit(16단어)블로크안에서 통보문의 처리.** 알고리듬의기본은 16개 단계의 매개 처리의 10개 회전들로 이루어 진 모듈이다. 10개 회전을 5개 회전들의 두 병렬선으로 배치한다. 론리를 그림 9-8에서 보여 준다. 10개 회전은 류사한 구조를 가지지만 매개는 f1, f2, f3, f4 및 f5라고 하는 서로 다른 원시론리함수를 리용한다. 함수들을 오른쪽선에서 거꾸로 리용한다는것을 강조한다.

  매 회전은 입력으로서 (Y_q)를 처리하는 현재 512bit블로크와 160bit완충기값 ABCDE(왼쪽 선)또는 A′B′C′D′E′(오른쪽 선)를 취하여 완충기의 내용을 갱신한다. 매 회전은 또한 보충적인 상수를 리용한다. 사실 오직 9개의 서로 다른 상수들만을 리용하는데 매개는 령이다. 16진과 10진으로 그 값을 표 9-3에서 보여준다.

  CV_{q+1}를 생성해 내는 첫 회전(CV_q)에 대한 련쇄변수의 입력에 5번째 회전의(18번째 단계)출력을 첨가한다. 첨가된 mod 2^{32}에 관한 더하기를 리용하여 CV_q안의 매개 단어와 매선의 완충기에서 5개의 매개 단어에 대하여 독립적으로 한다. 첨가는 다음과 같은 하쉬함수에서 3개 입력의 매개 단어들의 회전을 포함한다.

$$CV_{q+1}(0) = CV_q(1) + C + D'$$
$$CV_{q+1}(1) = CV_q(2) + D + E'$$
$$CV_{q+1}(2) = CV_q(3) + E + A'$$
$$CV_{q+1}(3) = CV_q(4) + A + B'$$
$$CV_{q+1}(4) = CV_q(0) + B + C'$$

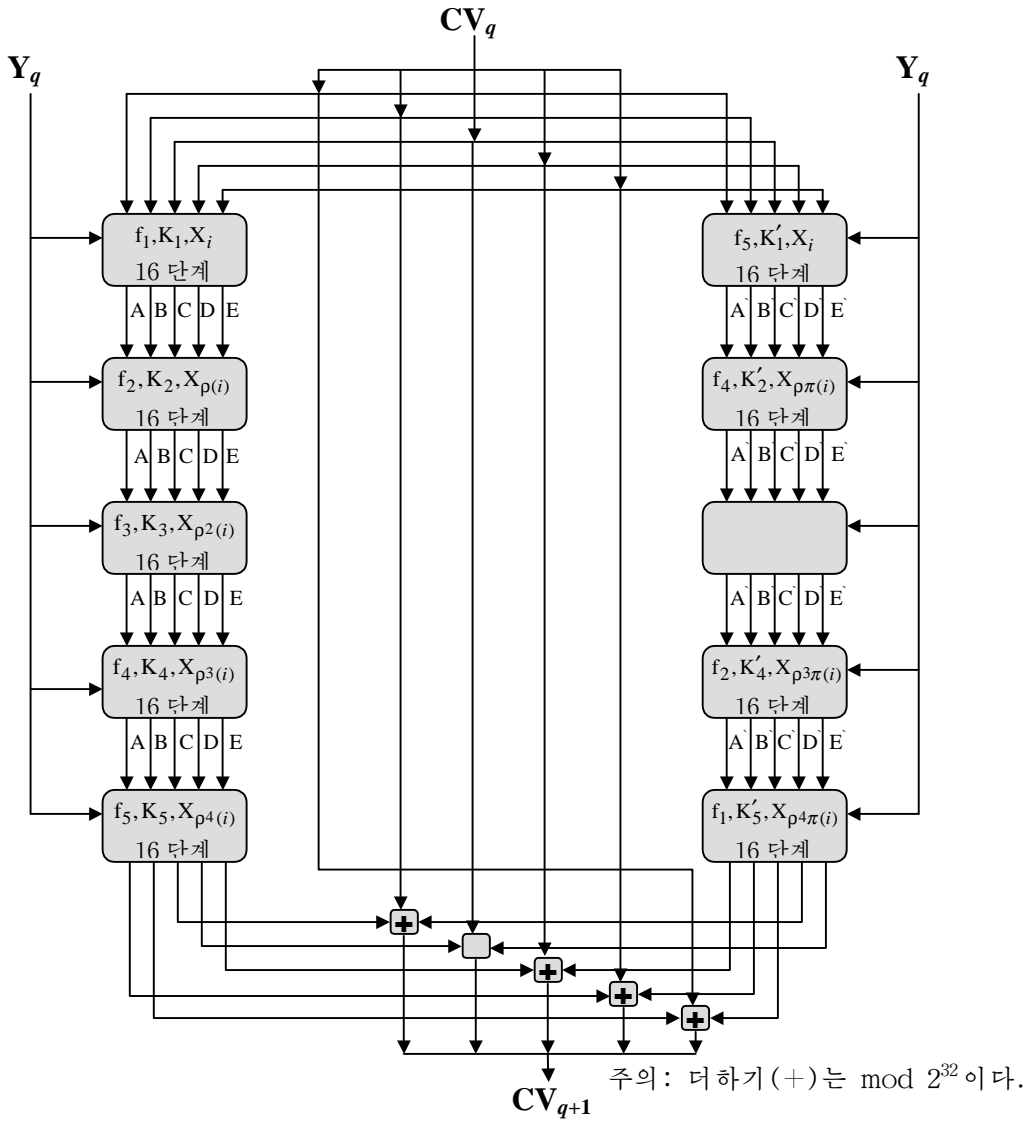



그림 9-8. 하나의 512bit블로크의 RIPEMD-160처리
(RIPEMD-160압축함수)

표 9-3. RIPEMD-160상수

구분	왼쪽절반		오른쪽절반	
단계수	16진	옹근수부	16진	옹근수부
$0 \le j \le 15$	$K_1=K(j)=$ 00000000	0	$K'_1=K'(j)=$ 50A28BE6	$2^{30} \times \sqrt[3]{2}$
$16 \le j \le 31$	$K_2=K(j)=$ 5A827999	$2^{30} \times \sqrt{2}$	$K'_2=K'(j)=$ 5C4DD124	$2^{30} \times \sqrt[3]{3}$
$32 \le j \le 47$	$K_3=K(j)=$ 6ED9EBA1	$2^{30} \times \sqrt{3}$	$K'_3=K'(j)=$ 6D703EF3	$2^{30} \times \sqrt[3]{5}$
$48 \le j \le 63$	$K_4=K(j)=$ 8F1BBCDC	$2^{30} \times \sqrt{5}$	$K'_4=K'(j)=$ 7A6D76E9	$2^{30} \times \sqrt[3]{7}$
$64 \le j \le 79$	$K_5=K(j)=$ A953FD4E	$2^{30} \times \sqrt{7}$	$K'_5=K'(j)=$ 00000000	0

- **단계5: 출력.** 모든 L개 512bit블로크를 처리한 다음에 L번째 계단으로부터의 출력은 160bit통보문요약이다.

RIPEMD-160 압축함수

하나의 512bit블로크처리의 10개의 매개 회전에서 론리를 좀 더 구체적으로 보자. 매 회전은 16개 단계들의 렬로 이루어 진다. 그림 9-9는 단계연산을 보여 준다. 5개 단어(A, B, C, D, E)를 리용하는 순서는 매 단계에 대한 하나의 단어의 단어준위오른쪽순환밀기를 생성해 낸다. 이 구조는 MD5의 구조와 거의 같다.

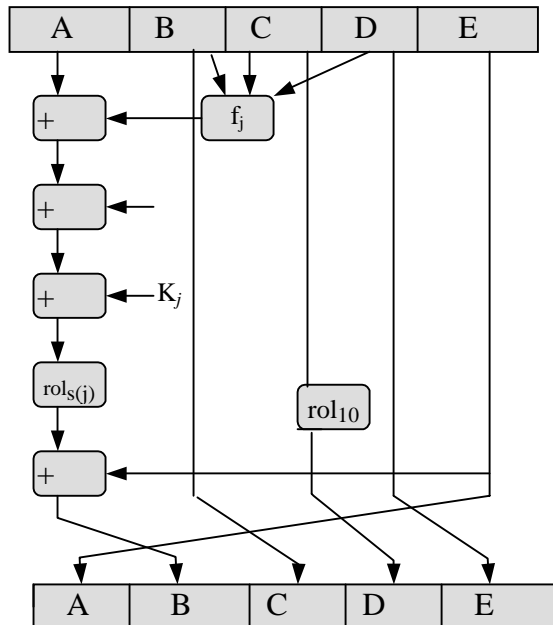



그림 9-9. 초등적인 RIPEMD-160연산(하나의 단계)

5개의 원시론리함수들중의 하나는 오른쪽 선우에서 거꿀순서로 리용된 함수들을 가진 알고리듬의 5개의 매개 회전에 리용한다(그림 9-8). 매 원시함수는 3개의 32bit단어를 입력으로 취하여 32bit단어를 출력으로 내보낸다. 매 함수는 비트별론리연산들의 모임을 진행한다. 즉 출력의 n번째 비트는 3개 입력의 n번째 비트에 관한 함수이다. 함수를 다음과 같이 개괄할수 있다.

단계	함수이름	함수값
$0 \le j \le 15$	$f_1=f(j,B,C,D)$	$B \otimes C \otimes D$
$16 \le j \le 31$	$f_2=f(j,B,C,D)$	$(B\wedge C)\vee(\neg B\wedge D)$
$32 \le j \le 47$	$f_3=f(j,B,C,D)$	$(B\wedge D)\vee(C\wedge\neg D)$
$48 \le j \le 63$	$f_4=f(j,B,C,D)$	$B \oplus (C\vee\neg D)$
$64 \le j \le 79$	$f_5=f(j,B,C,D)$	$B \otimes (C\vee D)$

론리연산자(AND, OR, NOT, XOR)들을 각각 기호($\wedge, \vee, \overline{}, \oplus$)로 표현한다. 함수 f_1는 기우성비트를 내보낸다. 함수 f_2은 조건부함수이다. 즉 B이면 C이고 아니면 D이다. 마찬가지로 f_4를 다음과 같이 정식화할수 있다. D이면 B이고 아니면 C이다. 표 9-4는 5개 함수의 진리표이다.

표 9-4. RIPEMD-160에 대한 론리함수의 진리표

B	C	D	f1	f2	f3	f4	f5
0	0	0	0	0	1	0	1
0	0	1	1	1	0	0	0
0	1	0	1	0	0	1	1
0	1	1	0	1	1	0	1
1	0	0	1	0	1	0	0
1	0	1	0	0	0	1	1
1	1	0	0	1	1	1	0
1	1	1	1	1	0	1	0

문헌[DOBB96b]로부터 채용한 다음의 준부호는 하나의 회전에 대한 처리알고리듬을 정의한다.

$A:=CV_q(0);B:=CV_q(1);C:=CV_q(2);D:=CV_q(3);E:=CV_q(4);$
$A':=CV_q(0);B':=CV_q(1);C':=CV_q(2);D':=CV_q(3);E':=CV_q(4);$
for j:=0 to 79 do
  $T:=rol_{s(j)}(A+f(j,B,C,D)+X_{r(j)}+K(j))+E;$
  $A:=E;E:=D;D:=rol_{10}(C);C:=B;B:=T;$
  $T:=rol_{s'(j)}(A'+f(79-j,B',C',D')+X_{r'(j)}+K'(j))+E';$
  $A':=E';E':=D';D':=rol_{10}(C');C':=B';B':=T';$
enddo
$CV_{q+1}(0)=CV_q(1)+C+D';CV_{q+1}(1)=CV_q(2)+D+E';CV_{q+1}(2)=CV_q(3)+E+A';$
$CV_{q+1}(3)=CV_q(4)+A+B';\ CV_{q+1}(4)=CV_q(0)+B+C';$

여기서

A, B, C, D, E	:왼쪽 선에 대한 완충기의 5개 단어
A′, B′, C′, D′, E′	:오른쪽 선에 대한 완충기의 5개 단어
j	:단계수, $0 \le j \le 79$
$f(j,B,C,D)$	:왼쪽 선의 단계 j와 오른쪽 선의 단계 79-j에 리용된 원시론리함수
$rol_{s(j)}$	:32bit인자의 왼쪽 순환밀기(회전)이고 $s(j)$는 특별한 단계에 대한 회전량을 결정하는 함수이다.
$X_{r(j)}$	:현재 512bit입력블로크로부터의 32bit단어이고 $r(j)$는 특별한 단어를 선택하는 치환함수이다.
$K(j)$	:단계 j에서 리용된 첨가상수
+	:mod 2^{32}에 관한 더하기

32bit단어들의 배렬 X[0..15]는 처리할 현재의 512bit입력블로크의 값을 취한다. 한 회전내에서 16개 단어들의 매 $X[i]$를 매 선우에서 한단계동안에 정확히 두번 리용한다. 그리고 이 단어들을 리용하는 순서는 회전에 따라 변한다. 표 9-5의 ㄱ은 매선에서 매 회전에 리용된 치환을 보여 준다. 치환 π를 $\pi(i)=9i+5(\text{mod } 16)$으로 표현할수 있다. 표 9-5의 ㄴ은 매 회전에서 리용된 왼쪽순환밀기를 정의한다. 표는 특별한 32bit단어를 처리할 때 단계에 리용된 밀기의 총 량을 보여 준다. 이것은 밀기를 리용하는 순서가 아니다. 그 순서는 단어를 리용할 순서에 관계된다.

RIPEMD-160 설계결정

여기서는 강한 암호학적하쉬함수를 설계하는데서 고려되여야 하는 구체적인 수준의 몇가지 사상을 얻기 위한 RIPEMD-160의 개발자들에 의해 제작된 몇가지 설계결정을 보기로 한다[DOBB96a].

1. 다섯개의 매개 회전의 두 방향선을 리용하여 회전들사이의 충돌구하기의 복잡성을 증가시키는데 이것을 압축함수의 충돌을 구하기 위한 시작점으로 리용할수 있다.
2. 단순성을 위하여 두 선은 본질적으로 같은 론리를 리용하거나 설계자들은 가능한껏 두 선사이에 많은 차이를 도입하는것이 필요하다는것을 느끼였다. 설계자들은 가까운 미래에 두 선중의 하나와 두 병렬선의 세개 회전까지를 공격하는것이 가능하지만 그 차이때문에 두 평행선의 결합이 공격을 방지한다는것을 관찰하였다. 그 차이는 다음과 같다.
   ㄱ) 두선의 보충적인 상수들은 서로 다르다(표 9-3).
   ㄴ) 원시론리함수(f_1부터 f_5까지)들의 순서는 반전된다.
   ㄷ) 통보문블로크에서 32bit단어들의 처리순서는 서로 다르다(표 9-5의 ㄱ).
3. 4개 단어가 아니라 5개 단어를 리용하는것을 제외하고는 RIPEMD-160을 위한 단계연산은 하나의 보충적인 제외 즉 10bit위치에 의한 C단어의 회전을 가지는 MD5(그림 9-3)와 동일하다. 이 회전은 가장 유효한 비트에 집중하는 MD5공격을 피한다(문헌[BOER93]). 값 10은 다른 회전들에 리용되지 않으므로 그것을 선택하였다.
4. 치환 π(표 9-5의 ㄱ)는 한 회전내에서 가까운 두 통보문단어가 다음번에는 상대적으로 멀리 떨어 지게 하는 효과를 가진다. 치환 π(표 9-5의 ㄱ)는 왼쪽 선에서 가까운 두 통보문단어는 항상 오른쪽 선에서 적어도 7개 위치가 떨어 져

있도록 선택되였다.

5. 왼쪽순환밀기(표 9-5의 ㄴ)는 다음과 같은 설계판정기준에 기초하여 선택되였다.
   ㄱ) 밀기는 5부터 15사이의 범위이다. 5보다 적은 밀기는 약한것으로 고찰되였다.
   ㄴ) 매 통보문단어는 같은 기우성을 가지는 일부 5개 회전에 대한 서로 다른 량쪽우에서 회전된다.
   ㄷ) 매 단어에 적용된 밀기는 특수한 패턴을 가지지 않을것이다(즉 총적인것은 32로 나누어 지지 말아야 할것이다).
   ㄹ) 너무 많은 밀기상수들은 4로 나누어 지지 않을것이다.

MD5와 SHA-1과의 비교

RIPEMD-160은 많은 측면에서 MD5와 SHA-1과 류사하다. 왜냐하면 이 세 알고리듬모두는 MD4로부터 유도된다고 기대되기때문이다. 표 9-6은 몇가지 류사성과 차이점을 보여 준다. 여러가지 관점에서 다음과 같이 설계들에 설명을 줄수 있다.

표 9-5. RIPEMD-160의 요소

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\rho(i)$	7	4	13	1	10	6	15	3	12	0	9	5	2	14	11	8
$\pi(i)$	5	14	7	0	9	2	11	4	13	6	15	8	1	10	3	12

선	1회전	2회전	3회전	4회전	5회전
왼쪽	단위치환	ρ	ρ^2	ρ^3	ρ^4
오른쪽	π	$\rho\pi$	$\rho^2\pi$	$\rho^3\pi$	$\rho^4\pi$

ㄱ) 통보문단어들의 치환

회전	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
1	11	14	15	12	5	8	7	9	11	13	14	15	6	7	9	8
2	12	13	11	15	6	9	9	7	12	15	11	13	7	8	7	7
3	13	15	14	11	7	7	6	8	13	14	13	12	5	5	6	9
4	14	11	12	14	8	6	5	5	14	12	15	14	9	9	8	6
5	15	12	13	13	9	5	8	6	15	11	12	11	8	6	5	5

ㄴ) 통보문단어들의 왼쪽순환밀기(두개 선 다)

- **힘내기공격에 대한 방지:** 세 알고리듬모두는 약충돌방지에 대한 공격에 약하지 않다. 128bit에서 MD5는 강충돌방지에서 생일공격에 아주 약하다면 SHA-1과 RIPEMD-160은 둘 다 가까운 장래에는 안전하다.
- **암호분석에 대한 방지:** 앞에서 론의한바와 같이 MD5의 암호분석에서는 많은 전진이 이루어 졌다. RIPEMD-160은 특히 알려 진 암호분석공격을 방지하기 위하여 설계되였다. 비록 SHA-1의 설계원리를 대략 조금 알고 있다고 할지라도 알려 진 암호분석공격들을 강하게 방지하는것이 가능하다. 진행된 단계수를 반복하는 두 처리사이의 리용은 SHA-1에 비하여 보다 힘든 암호분석이 되도록 복잡성을 첨가한 RIPEMD-160을 준다.

표 9-6. MD5,SHA-1과 RIPEMD-160의 비교

	MD5	SHA-1	RIPEMD-160
요약정보길이	128bit	160bit	160bit
처리의 기본단위	512bit	512bit	512bit
단계의 수	64(16의 4회전)	80(20의 4회전)	160(16의 5개쌍회전)
최대통보문크기	∞	2^{64}-1bit	2^{64}-1bit
원시론리함수	4	4	5
리용된 보충상수	64	4	9
Endianness	Little-endian	Big-endian	Little-endian

몇가지 하쉬함수의 상대적성능
표 9-7. (266MHz Pentium우에서 C++로 작성한)

알고리듬	Mbps
MD5	32.4
SHA-1	14.4
RIPEMD-160	13.6

- **속도:** 세개의 알고리듬모두는 mod 2^{32}에 관한 더하기와 단순한 비트별 론리연산에 의거하는데 이 모두는 32bit방식에서 잘 진행된다. 첨가된 복잡성과 SHA-1 및 RIPEMD-160의 단계수는 MD5에 비하여 느리게 비례한다. 표 9-7은 266MHz Pentium우에서 달성된 일련의 결과들을 보여준다. 류사한 상대적결과들을 문헌[BOSS96]에 주었다.
- **Little-endian 대 Big-endian방식:** MD5와 RIPEMD-160은 32bit단어들의 렬로서 통보문을 해석하는 little-endian방식을 리용한다면 SHA-1은 big-endian방식을 리용한다. 어느 방식도 우월한것은 없다.

9.4 HMAC

8장에서 대칭블로크암호의 리용에 기초한 통보문인증부호(MAC) 다시말하여 FIPS PUB 113에서 정의된 자료인증알고리듬의 실례를 보았다. 이것은 전통적으로 MAC를 구성하는 가장 공통적인 방식이였다. 최근년간에는 암호학적하쉬부호로부터 유도된 MAC를 발전시키는데 관심이 높아 졌다. 관심이 높아 진 동기는 다음과 같다.

1. MD5와 SHA-1과 같은 암호학적하쉬함수는 일반적으로 DES와 같은 대칭블로크암호보다 쏘프트웨어적으로 고속으로 실행된다.
2. 암호학적하쉬함수의 서고부호를 널리 보급한다.
3. 암호학적하쉬함수에 대하여 미국 또는 다른 나라에서 수출제한이 없다. MAC로 리용될 때도 대칭블로크암호는 제한되고 있다.

MD5와 같은 하쉬함수는 MAC로서 리용하기 위해 설계되지 않았으며 따라서 그 목적을 위해 직접 리용할수는 없다. 왜냐하면 비밀열쇠에 의거하지 않기때문이다. 존재하는 하쉬알고리듬에 비밀열쇠의 통합을 위한 일련의 제의들도 있었다. 최대의 지지를 얻

은 방식은 HMAC이다(문헌[BELL96a,BELL96b]). HMAC는 RFC2104로서 발행되였고 IP보안을 위한 명령실현 MAC를 선택하였으며 SSL과 같은 다른 인터네트규약으로 리용되고 있다.

HMAC 설계대상

RFC2104는 HMAC을 위한 다음과 같은 설계대상들을 렬거한다.

- 변경없이 리용하기 위하여 하쉬함수를 허용한다. 특히 쏘프트웨어적으로 잘 진행되며 부호가 자유롭고 널리 허용되는 하쉬함수
- 매장된 하쉬함수의 쉬운 재배치성을 허용하기 위하여 고속 또는 안전한 하쉬함수들이 경우에 따라 발견되거나 요구된다.
- 유효한 회화를 초래함이 없이 하쉬함수의 원래의 성능을 보존하기 위하여
- 간단한 방법으로 열쇠를 리용하고 조종하기 위하여
- 매장된 하쉬함수에 대한 적당한 가정에 기초하여 인증기구의 강도가 잘 리해되는 암호학적분석을 가지기 위하여

첫 두 대상은 HMAC의 접수성에서 중요하다. HMAC는 하쉬함수를 《검은 통》으로 취급한다. 이것은 2개의 우점을 가진다. 우선 하쉬함수의 존재하는 실현을 HMAC를 실현하는데서 모듈로 리용할수 있다. 이 방법에서 HMAC부호의 크기는 미리 준비되여 있어 변경없이 리용하기 쉽다. 다음으로 HMAC실현에서 주어 진 하쉬함수를 재배치하기를 바란다면 그모두는 존재하는 하쉬함수모듈을 제거하고 새로운 모듈을 받아 들이는것이다. 이것은 고속하쉬함수를 요구할 때 진행된다. 보다 중요하게는 매장된 하쉬함수의 보안이 손상되였다면 HMAC의 보안은 매장된 하쉬함수를 보다 안전한것으로 재배치함으로써(즉 MD5를 RIPEMD-160으로 교체함으로써) 단순하게 유지될수 있는것이다.

우의 렬거에서 마지막설계대상은 사실상 다른 제기된 하쉬기초방식우에서 HMAC의 주요우점이다. 매장된 하쉬함수가 적당한 암호학적강도를 가진다는 전제하에서 HMAC를 안전하게 증명할수 있다. 이 절에서는 후에 이 점을 취급하지만 우선 HMAC의 구조를 설명한다.

HMAC 알고리듬

그림 9-10은 HMAC의 전면적인 조작을 보여 준다. 다음과 같은 표시들을 정의한다.

H: 매장된 하쉬함수(즉 MD5, SHA-1, RIPEMD-160)
M: HMAC에로의 통보문입력(매장된 하쉬함수안에서 서술된 메꾸기를 포함)
Y_i: M의 i번째 블로크, $0 \le i \le L-1$
L: M안에서 블로크수
b: 블로크안에서 비트수
n: 매장된 하쉬함수에 의해 생성된 하쉬부호의 길이
K: 비밀열쇠로서 열쇠길이가 b보다 크면 열쇠는 n bit열쇠를 생성해 내는 하쉬함수에로의 입력이며 요구되는 길이는 $\ge n$이다.
K^+: 결과가 b bit가 되도록 왼쪽에 령을 메꾼 K
ipad: b/8번 반복된 00110110
opad: b/8번 반복된 01011010

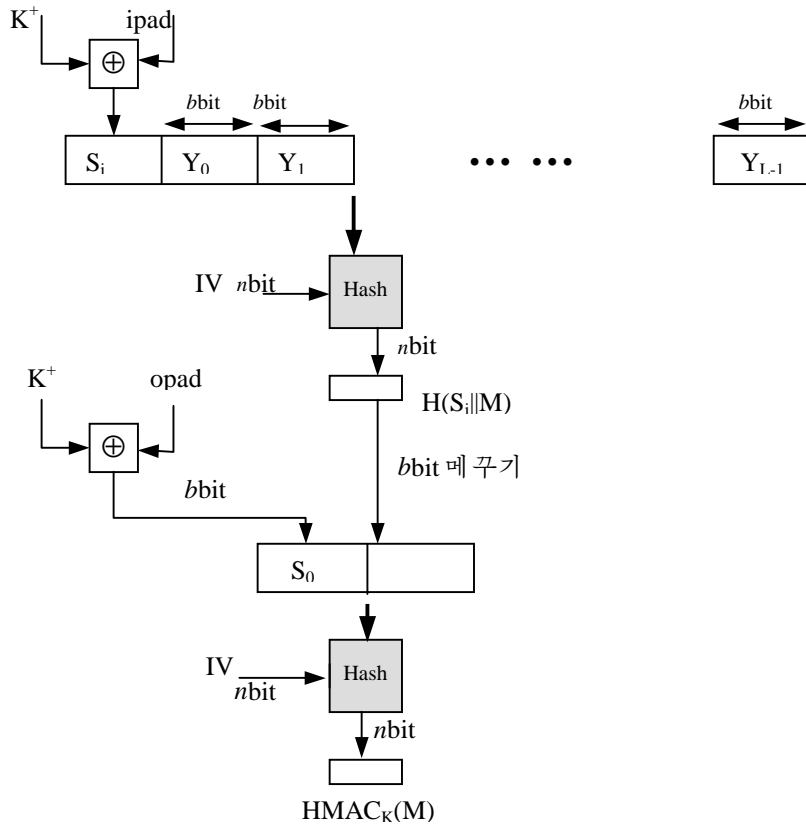



그림 9-10. HMAC구조

이때 HMAC를 다음과 같이 표현할수 있다.

$$HMAC_K = H[(K^+ \oplus opad) \| h[(K^+ \oplus ipad) \| M]$$

달리 말하여

1. K의 왼쪽 끝에 령을 첨가하여 bbit렬 K^+를 창조한다(즉 K가 길이 160bit이고 b=512이면 K는 44개 령바이트 Ox00으로 보충될것이다).
2. K^+와 opad를 배타적론리합하여 bbit블로크 S_i를 생성해 낸다.
3. M을 S_i에 첨가한다.
4. H를 단계 3에서 생성된 흐름에 적용한다.
5. K^+와 opad를 배타적론리합하여 $b$bit블로크 S_0을 생성해 낸다.
6. 단계 4부터의 하쉬결과를 S_0에 첨가한다.
7. H를 단계 6에서 생성된 흐름에 적용하여 결과를 출력한다.

ipad와의 배타적론리합은 K의 비트들을 절반으로 쪼개여 나타난다는것을 강조한다. 마찬가지로 opad와의 배타적론리합은 K의 비트들을 절반으로 쪼개여 나타나지만 비트

들의 모임은 차이난다. 사실 하쉬알고리듬의 압축함수를 통하여 S_i와 S_0을 보냄으로써 모조란수적으로 K로부터 두 열쇠를 생성하였다.

HMAC는 근사적으로 이 통보문에 대한 매장된 하쉬함수와 같은 시간을 실행한다. HMAC는 하쉬압축함수의 3개 실행을 첨가한다(S_0, S_i와 내부하쉬로부터 생성된 블로크에 대하여).

보다 효률적인 실현이 가능한데 그림 9-11에서 보여 준것과 같다. 다음과 같은 2개 값을 미리 계산한다.

$$f(IV, (K^{+} \oplus ipad))$$
$$f(IV, (K^{+} \oplus opad))$$

여기서 f(cv, block)는 하쉬함수에 대한 압축함수로서 *n* bit의 련쇄변수와 *b* bit의 블로크를 인수로 취하여 *n* bit의 련쇄변수를 내보낸다. 이 값들은 오직 초기에 그리고 매번 열쇠변화를 계산할것을 요구한다. 사실상 미리 계산된 값들은 하쉬함수에서의 초기값(IV)

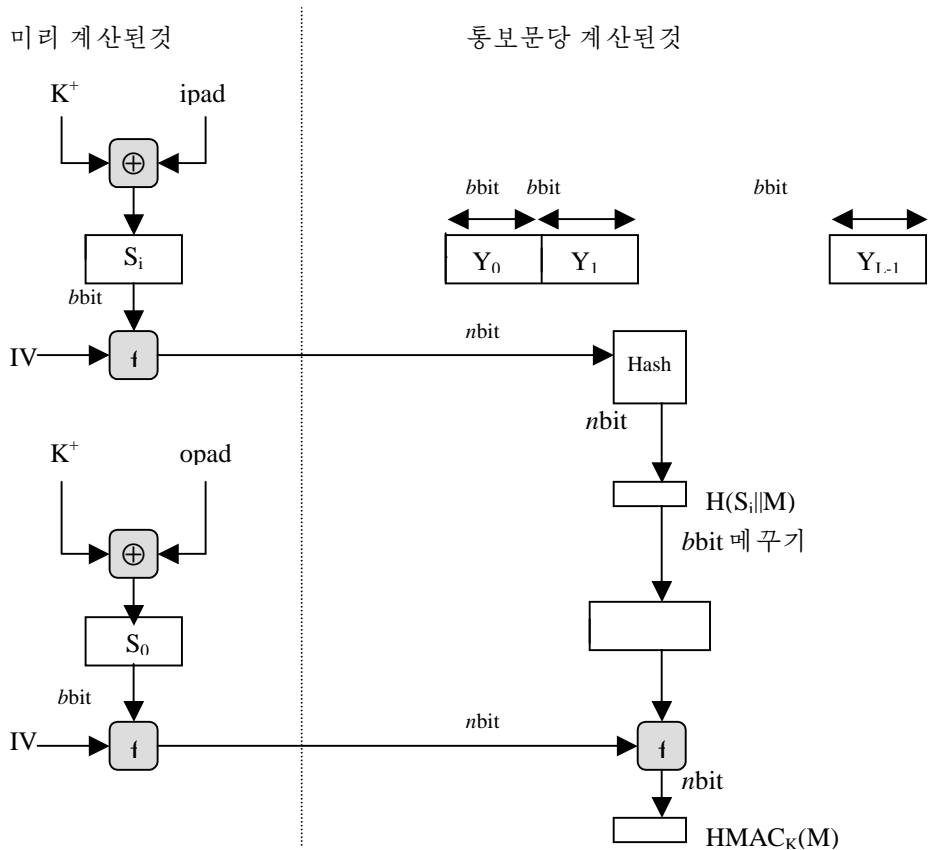



그림 9-11. HMAC의 효과적인 실현

으로 치환한다. 이 실현은 오직 하나의 압축함수의 보충적인 실체를 하쉬함수에 의해 표준적으로 생성된 처리에 첨가한다. 이보다 효과적인 실현은 MAC를 계산하는 대부분의 통보문들이 짧다면 특별히 가치가 있다.

HMAC의 보안

매장된 하쉬함수에 기초한 임의의 함수의 보안은 기본적인 하쉬함수의 암호학적강도에 대한 몇가지 방법에 의거한다. HMAC의 흥미를 끌게 한것은 그 설계가들이 HMAC의 강도와 매장된 하쉬함수의 강도사이의 정확한 관계를 증명할수 있었다는것이다.

MAC함수의 보안은 일반적으로 위조자와 같은 열쇠로 창조된 통보문-MAC쌍의 주어 진 수에 의해 소비된 주어 진 시간량으로 성과적인 위조확률에 의해 표현된다. 본질에 있어서 주어 진 효과수준(시간, 통보문-MAC쌍)에 대하여 정당한 사용자와 공격자에의하여 HMAC에 대한 성공적인 공격의 확률은 매장된 하쉬함수에 대한 다음과 같은 공격들중의 하나와 동등하다.

1. 공격자는 우연적이고 안전(비밀)하며 공격자에게는 알려 지지 않은 IV와 압축함수의 출력을 계산할수 있다.
2. 공격자는 IV가 우연적이고 안전(비밀)할 때 하쉬함수에서 충돌을 구한다.

첫번째 공격에서는 압축함수를 하나의 b bit블로크로 이루어 진 통보문에 적용된 하쉬함수와 동등한것으로 개괄할수 있다. 이 공격에 대하여 하쉬함수의 IV는 비밀이며 우연인 n bit값으로 교체된다. 이 하쉬함수에 대한 공격은 2^n준위의 효과적인 열쇠에 대한 힘내기공격이나 다음에 론의하는 두번째 공격의 특수경우인 생일공격을 요구한다.

두번째 공격에서는 공격자가 같은 하쉬 즉 H(M)＝H(M')를 내보내는 두 통보문 M과 M'를 보고 있다. 이것이 8장에서 론의한 생일공격이다. 이것은 n의 하쉬길이에 대하여 $2^{n/2}$의 효과준위를 요구한다는것을 보여 준다. 이 기초에서 MD5의 보안은 질문이 제기된다. 왜냐하면 2^{64}효과준위는 오늘날 기술적으로 실행가능하기때문이다. 이것은 MD5와 같은 128bit하쉬함수는 HMAC에 대하여 적합치 않다는것을 의미하는가? 그에 대한 대답은 부정적이다. 왜냐하면 다음과 같은 론의때문이다. MD5를 공격하기 위하여 공격자는 임의의 통보문들의 모임을 선택하고 비직결적으로 충돌을 구하는 편의를 계산하는 로출된 상태우에서 이것들을 동작시킨다. 공격자는 하쉬알고리듬과 음적 IV를 알고 있으므로 자기가 생성하는 매 통보문에 대한 하쉬부호를 생성할수 있다. 그러나 HMAC를 공격할 때 공격자는 통보문/부호 쌍을 비직결적으로 생성할수 없다. 왜냐하면 공격자는 K를 모르기때문이다. 그러므로 공격자는 같은 열쇠하에서 HMAC에 의해 생성된 통보문들의 렬을 관찰해야 하며 이 알려 진 통보문우에서 공격을 진행해야 한다. 128bit의 하쉬부호길이에 대하여 이것은 같은 열쇠를 리용하여 생성된 2^{64}개의 관찰된 블로크(2^{73}bit)를 요구한다. 1Gbps련결에서는 성공을 위해 약 250000년동안 열쇠에서의 변화없이 련속통보문흐름을 관찰해야 한다. 이리하여 속도에 관계된다면 HMAC에 대한 매장된 하쉬함수로서 SHA-1이나 RIPEMD-160보다도 오히려 MD5를 리용하는것이 완전히 접수가능하다.

문 제

1. MD5와 MD4사이의 차이를 설명하라. 특히 MD5가 MD4보다 더 강하다고 생각되는 것은 무엇이며 왜 그런가?
2. 그림 9-7에서는 80개의 32bit단어들의 배렬이 W_t의 값을 기억할수 있다고 가정한다. 따라서 블로크처리의 시작에서 미리 계산할수 있다. 이제 공간이 상금에 있다고 가정한다. 다른 하나의 방식으로서 초기에 W_0부터 W_{15}까지를 적재하는 16bit단어순환완충기의 리용을 고찰하자. 매 단계 t에 대하여 요구된 초기값 W_t를 계산하는 알고리듬을 설계하시오.
3. SHA-1에 대하여 W_{16} ,W_{17}, W_{18}, W_{19}의 값을 보여 주시오.
4. $a_1a_2a_3a_4$가 32bit단어에서의 4개 바이트이라고 가정하자. 매 요소를 2진으로 표현된 0부터 255사이의 옹근수로 고찰할수 있다. Big-endian방식에서 이 단어는 옹근수

$$a_1 2^{24} + a_2 2^{16} + a_3 2^8 + a_4$$

를 표현한다. Little-endian방식에서 이 단어는 옹근수

$$a_4 2^{24} + a_3 2^{16} + a_2 2^8 + a_1$$

를 표현한다.

ㄱ) MD5와 RIPEMD-160은 little-endian방식이라고 가정한다. 통보문요약이 기본적인 방식과 독립이라는것이 중요하다. 그러므로 big-endian방식우에서 mod 2에 관한 MD5나 RIPEMD-160의 더하기연산을 진행하기 위하여서는 조종해야 한다. $X=x_1x_2x_3x_4$이고 $Y=y_1y_2y_3y_4$라고 가정하자. MD5의 더하기연산($X+Y$)가 big-endian기계우에서 어떻게 수행되는가를 보여 주시오.

ㄴ) SHA-1은 big-endian방식이라고 가정하자. SHA-1에 대하여 연산($X+Y$)가 little-endian기계우에서 어떻게 수행되는가를 보여 주시오.

제10장. 수자서명과 인증규약

공개열쇠암호학에 관한 작업에서 가장 중요한 발전은 수자서명이다. 수자서명은 임의의 다른 방법으로 실현하는것이 힘든 보안능력들의 모임을 제공한다. 이 장에서는 수자서명에 대하여 개괄한다. 다음으로 수자서명의 리용에 의거한 인증규약을 본다. 최종적으로 수자서명표준(DSS)을 도입한다.

10.1 수자서명

요구

통보문인증은 임의의 제3자로부터 통보문을 교환하는 두 측을 보호한다. 그러나 호상간에 두 측을 보호하지는 못한다. 둘사이의 몇가지 형태의 론쟁(dispute)이 가능하다.

실례로 존(John)이 그림 8-4의 방식들중의 하나를 리용하여 인증된 통보문을 마리(Mary)에게 보낸다고 가정하자. 다음과 같은 론쟁을 고찰하자.

1. 마리가 다른 통보문을 위조하여 그것이 존으로부터 왔다고 고집할수는 있다. 존과 마리가 공유한 열쇠를 리용하여 간단히 통보문을 창조하여 인증부호를 첨가해야 한다.
2. 존은 통보문보내기를 부인할수 있다. 마리가 통보문을 위조할수 있으므로 존이 사실상 통보문을 보냈다는것을 증명할 방도가 없다.

두 씨나리오는 정당한 관계에 있다. 여기에는 첫 씨나리오의 실례가 있다. 전자자금조달자를 취하여 수신자는 전달된 자금의 량을 증가시키고 많은 자금이 송신자로부터 보내왔다고 주장한다. 두번째 씨나리오의 실례로 전자우편통보문에 나쁘게 전환된 업무를 처리하기 위한 주식중매인에 대한 지시를 포함하는것이다. 송신자는 통보문을 보낸일이 없다고 속인다.

송수신자사이의 완전한 신용이 없는 환경하에서는 인증보다 더 좋은것이 요구된다. 이 문제에 대한 가장 매력적인 풀이는 수자서명이다. 수자서명은 손으로 쓴 수표와 류사하다. 그것은 다음과 같은 성질들을 가져야 한다.

- 서명한 당사자와 날자 그리고 시간을 검증할수 있어야 한다.
- 서명시각에 내용이 인증될수 있어야 한다.
- 서명은 론쟁을 재풀기 위하여 제3자에 의해 검증가능해야 한다.

이리하여 수자서명함수는 인증함수를 포함한다.

이 성질에 토대하여 수자서명을 위해 다음과 같은 요구를 정식화할수 있다.

- 서명은 서명할 통보문에 관계되는 비트패턴이여야 한다.
- 서명은 위조와 부인을 둘 다 방지하기 위하여 송신자에 대한 유일한 어떤 정보를 리용해야 한다.
- 수자서명을 식별하고 검증하는것은 상대적으로 쉬워야 한다.
- 존재하는 수자서명에 대해 새로운 통보문을 구성하거나 주어 진 통보문에 대해 부정적인 수자서명을 구성함으로써 수자서명을 위조하는것이 계산량적으로 불가능해야 한다.
- 기억된 수자서명의 복사를 유지하는것은 실천적이여야 한다.

그림 8-5의 ㄷ 또는 ㄹ과 같은 방식으로 매장된 안전한 하쉬함수는 이 요구들을 만족시킨다.
수자서명함수에 대하여 한가지 변종의 방식을 제기하였다. 이 방식은 두가지 부류 즉 직접과 중재에로 귀착된다.

직접수자서명

직접수자서명은 통신측(원천지, 목적지)들만을 포함한다. 목적지에서는 원천지의 공개열쇠를 안다고 가정한다. 수자서명은 송신자의 비밀열쇠(그림 8-1의 ㄷ)로 전체적인 통보문을 암호화하거나 송신자의 비밀열쇠(그림 8-5의 ㄷ)로 통보문의 하쉬부호를 암호화함으로써 이루어 질수 있다.

수신자의 공개열쇠(공개열쇠암호)나 공유한 비밀열쇠(전통암호)로 전체적인 통보문+서명을 암호화함으로써 기밀성을 담보할수 있다. 실례로 그림 8-1의 ㄹ과 8-5의 ㄹ을 보시오. 우선 서명함수를 진행하고 그다음에 바깥기밀성함수를 진행하는것이 중요하다는 것을 강조한다. 론쟁의 경우에 어떤 제3자가 통보문과 그 서명을 보아야 한다. 서명을 암호화된 통보문으로 계산하였다면 제3자에게는 또한 원래의 통보문을 읽기 위해 복호화열쇠를 참조할것을 요구한다. 그러나 서명이 내부연산이면 수신자는 후에 론쟁해결에 리용하기 위하여 평문통보문과 그 서명을 기억시킬수 있다.

지금까지 고찰한 모든 직접방식들은 공통적인 약점을 가진다. 즉 방식의 타당성은 송신자의 비밀열쇠의 보안에 의거한다. 송신자가 후에 특별한 통보문보내기를 부인하려고 한다면 송신자는 자기의 비밀열쇠가 분실되였거나 도적맞히였으며 어떤 사람이 자기의 서명을 위조하였다고 고집할수 있다. 비밀의 보안에 관계되는 행정적인 조치를 채용하여 이런 현상을 방지하거나 적어도 약화시킬수는 있지만 위협은 아직 어느 정도로 남아 있다. 하나의 실례는 일부인(시간과 날자)을 포함하는 매 서명된 통보문을 요구하며 중심국에 손상된 열쇠를 신속히 보고할것를 요구하는것이다.

다른 하나의 위협은 시각 T에 X로부터 어떤 비밀열쇠가 실제적으로 도적 맞히는것이다. 이때 적은 X의 서명과 T이전시간에 도장을 찍은 서명된 통보문을 보낼수 있다.

중재수자서명

직접수자서명과 련관된 문제는 중재인을 리용하여 써놓을수 있는것이다.

직접수자서명에서처럼 중재서명방식의 변종이 있다. 일반항목에서 그것들은 모두 다음과 같은것을 고찰한다. 송신자 X로부터 수신자 Y에로의 매 서명된 통보문은 우선 중개자 A에게 가는데 중개자는 통보문과 그 서명을 원본과 내용을 검열하는 일련의 검사자의 역할을 한다. 이때 통보문에 날자를 써넣고 중개자의 만족을 표현한 내용과 함께 Y에게 보낸다. A의 출현은 직접수자서명방식에서 제기된 문제 즉 X가 통보문을 부인할

수 있는 문제를 푼다.

중개자는 이런 종류의 방식에서 신속하고 결정적인 역할을 하며 모든 측들은 중재기구가 적당히 동작한다는 신용이 있어야 한다. 16장에서 언급된 신용체계의 리용은 이 요구를 만족시킬수 있다.

문헌[AKL83]과 [MITC92]에서 언급된 씨나리오에 기초한 표 10-1은 중재수자서명의 몇가지 실례를 준다. 첫째로, 전통암호를 리용한다. 송신자 X와 중개자 A는 비밀열쇠 K_{xa}를 공유하며 A와 Y는 비밀열쇠 K_{ay}를 공유한다고 가정한다. X는 통보문 M을 구성하고 그의 하쉬부호 H(M)을 계산한다. 그다음 X는 통보문과 서명을 A에게 전송한다. 서명은 X의 식별자와 K_{xa}을 리용하여 모두 암호화된 하쉬값으로 이루어 진다. A는 서명을 복호화하고 통보문을 타당하게 하는 하쉬값을 검열한다. 그다음에 A는 K_{ay}로 암호화된 통보문을 Y에게 보낸다. 통보문은 ID_X, X로부터의 원래의 통보문, 서명 그리고 시간도장을 포함한다.

표 10-1. 중재수자서명기술

ㄱ) 전통암호, 중개자는 통보문을 본다.
① X→A: $M \| E_{K_{xa}}[ID_X \| H(M)]$
② A→Y: $E_{K_{ay}}[ID_X \| M \| E_{K_{xa}}[ID_X \| H(M)] \| T]$
ㄴ) 전통암호, 중개자는 통보문을 보지 못한다.
① X→A: $ID_X \| E_{K_{xy}}[M] \| E_{K_{xa}}[ID_X \| H(E_{K_{xy}}[M])]$
② A→Y: $E_{K_{ay}}[ID_X \| E_{K_{xy}}[M] \| E_{K_{xa}}[ID_X \| H(E_{K_{xy}}[M])] \| T]$
ㄷ) 공개열쇠암호,중개자는 통보문을 보지 못한다.
① X→A: $ID_X \| E_{KR_x}[ID_X \| E_{KU_y}(E_{KR_x}[M])]$
② A→Y: $E_{KR_a}[ID_X \| E_{KU_y}[E_{KR_x}[M]] \| T]$

주의: X-송신자, Y-수신자, A-중개자, M-통보문

Y는 이것을 복호화하여 통보문과 서명을 얻는다. 일부인은 이 통보문이 시기적절하며 재시합이 아님을 Y에게 통지한다. Y는 M과 서명을 기억할수 있다. 론쟁의 경우에 M을 X로부터 수신하였다고 고집하는 Y는 다음과 같은 통보문을 A에게 보낸다.

$$E_{K_{ay}}[ID_X \| M \| E_{K_{xa}}[ID_X \| H(M)]]$$

중개자는 $E_{K_{ay}}$ 를 리용하여 ID_X, M 그리고 서명을 얻으며 그다음에 $E_{K_{xa}}$ 를 리용하여 서명을 복호화하고 하쉬부호를 검증한다. 이 방식에서 Y는 직접 X의 서명을 검열할 수 없다. 서명은 오직 거기서 론쟁만을 해결할뿐이다. Y는 X로부터 인증된 통보문을 고찰한다. 왜냐하면 A를 통해서 오기때문이다. 이 씨나리오에서 두 측면은 A에서 높은 기밀성을 가져야 한다. 즉

- X는 K_{xa}가 드러나지 않으며 $E_{K_{xa}}[ID_X \| H(M)]$형태의 거짓서명을 생성하지 않는다고 A를 믿어야 한다.
- Y는 A를 믿고 하쉬값이 정확하며 서명이 X에 의해 생성되였을 때에만 $E_{K_{ay}}[ID_X \| M \| E_{K_{xa}}[ID_X \| H(M)] \| T]$를 보내야 한다.
- 두 측은 A가 론쟁을 공정하게 풀어 나간다고 믿어야 한다.

중개자가 이런 신용속에서 살면 X는 누구도 자기의 서명을 위조할수 없으며 Y는 X가 자기의 서명을 부인할수 없다는것을 담보 받는다.

앞의 씨나리오는 또한 A는 X에서 Y에로의 통보문을 읽을수 있으며 사실 임의의 도청자도 그렇게 할수 있다는것이 나온다. 표 10-1의 ㄴ은 사전에 중재를 제공하지만 역시 기밀성을 담보하는 씨나리오를 보여 준다. 이 경우에 X와 Y는 비밀열쇠 K_{xy}를 공유한다고 가정한다. 이제 X는 식별자, K_{xy}로 암호화된 통보문복사 그리고 서명을 A에게 전송한다. 서명은 식별자+모두 K_{xa}를 리용하여 암호화된 통보문의 하쉬값으로 이루어 진다. 사전에 A는 서명을 복호화하고 통보문을 타당하게 하는 하쉬값을 검열한다. 이 경우에 A는 오직 암호화된 통보문판과만 작업을 하며 그것을 읽는것으로부터 방지된다. A는 그다음에 X로부터 수신된 모든것+모두 K_{ay}로 암호화된 일부인을 Y에게 전송한다.

통보문을 읽을수 없다고 하더라도 중개자는 아직 X나 Y의 한 축우에서 협잡을 방지하는 위치에 있다. 나머지 문제인 첫번째 씨나리오로 공유한 대상은 중개자가 서명된 통보문을 부인하는 송신자 또는 송신자의 서명을 위조하는 수신자와 함께 동맹을 형성할수 있다는것이다.

론의된 모든 문제들은 공개된 방식으로 진행함으로써 해결할수 있는데 그 한가지를 표 10-1의 ㄷ에서 보여 준다. 이 경우에 X는 우선 X의 비밀열쇠 KR_x로, 그다음에는 Y의 공개열쇠 KU_y로 통보문 M을 2중암호화한다. 이것은 서명된 통보문의 비밀판이다. 이 서명된 통보문과 함께 X의 식별자는 다시 KR_x로 암호화되며 ID_X와 함께 A에게 보낸다. 2중암호화된 내부통보문은 중개자와 Y를 제외한 모두로부터 안전하다. 그러나 A는 통보문이 X로부터 와야 한다는것을 담보하는 바깥암호를 복호화한다(왜냐하면 오직 X만이 KR_x를 가지기때문이다). A는 X의 공개열쇠/비밀열쇠가 아직 타당하다는것을 믿도록 검열하며 그렇다면 통보문을 검증한다. 그다음에 A는 KR_a로 암호화된 통보문을 Y에게 전송한다. 통보문은 ID_X, 2중암호화된 통보문 그리고 일부인를 포함한다.

이 방식은 선행한 두 방식들에 비하여 일련의 우점을 가진다. 우선 속히워서 빼앗기는것을 방지하기 위해 통신하기전에 통신측들중에서 어떤 정보도 공유하는것이 없다. 둘째로 KR_x가 손상되였다고 할지라도 KR_a가 손상되지 않았다면 부정확한 날자가 찍힌 통보문을 보낼수 없다. 최종적으로 X에서 Y에로의 통보문내용은 A로부터 비밀이며 그 누구에게도 역시 그러하다.

10.2 인증규약

8장에서 서술된 기본도구는 10.1에서 론의한 수자서명을 포함하는 여러가지 응용에 리용된다. 다른 리용도 아주 많으며 계속 보충되고 있다. 이 절에서는 두개의 일반적인 분야(호상인증과 한방향인증)에 집중하며 이 두 분야에서의 인증의 몇가지 따름을 설명한다.

호상인증

중요한 인증분야는 호상인증규약이다. 이런 규약은 통신측들이 매개 다른 신원들을 호상 만족시키여 대화열쇠를 교환하게 할수 있게 한다. 이런 취지를 5.3(전통기술)과 6.3(공개열쇠기술)에서 설명하였다. 거기서 론의의 중심은 열쇠배포였다. 여기서는 보다 광범한 인증의 고찰에로 돌아 온다.

인증된 열쇠교환에서 중심문제는 두개의 론의점 즉 기밀성과 시기적절성(timeliness)이다. 대화열쇠의 가짜와 손상을 방지하기 위하여 본질적인 식별과 대화열쇠정보는 암호화된 형태로 통신되여야 한다. 이것은 이 목적을 위하여 리용될수 있는 선행한 비밀 또는 공개열쇠의 존재를 요구한다. 두번째 발행인 시기적절성은 통보문재시동의 위협때문에 중요하다. 이런 재시동은 최악의 경우에 대화열쇠를 손상시키거나 적이 다른 측을 성과적으로 재현하는것을 허용한다. 최소한도로 성과적인 재시동은 진짜인것처럼 보이지만 그렇지 않은 측들을 표현함으로써 연산을 분렬시킬수 있다.

문헌[GONG93]에서는 다음과 같은 재시동공격의 실례들을 렬거하였다.

- **단순재연:** 적은 단순히 통보문을 복사하여 그것을 후에 재연한다.
- **사용개시될수 있는 반복:** 적은 타당한 시간창문내에서 일부인된 통보문을 재연할 수 있다.
- **검출될수 없는 반복:** 원래의 통보문이 억제되여 그 목적지에 도착하지 못하고 아직 재연통보문만이 도착하므로 이런 상황이 발생한다.
- **변경없이 반대로의 재연:** 이것은 통보문송신자에게 반대로 재연하는것이다. 전통암호를 리용하고 송신자가 보낸 통보문과 내용의 기초우에서 수신된 통보문사이의 차이를 쉽게 식별할수 없으면 이 공격은 가능하다.

재연공격을 처리하는 한가지 방식은 인증교환에 리용되는 매 통보문에 렬번호를 붙이는것이다. 그 렬번호가 적당한 순서에 있을 때만 새 통보문을 접수한다. 이 방식의 어려움성은 취급된 매 청구자의 마지막렬번호를 따라 가는 모든 측을 요구하는것이다. 이 간접소비(overhead)때문에 렬번호는 일반적으로 인증과 열쇠교환에 리용되지 않는다. 그대신에 다음과 같은 두 방식중의 하나를 리용한다.

- **일부인:** A측은 통보문이 자기의 판단으로 자기의 현 시점의 지식에 충분히 가까운 일부인을 포함할 때만 통보문을 참신한것으로 접수한다. 이 방식은 각이한 참가자들이 박자를 동기화할것을 요구한다.
- **도전/응답:** B로부터 참신한 통보문이기를 기대하는 A측은 우선 B에게 림시적인것(도전)을 보내며 B로부터 수신된 그다음의 통보문(응답)은 정확한 림시적인 값을 포함할것을 요구한다.

일부인방 식은 이 기술의 고유한 어려움성으로 하여 련결지향응용에는 리용되지 않을것이라고 주장하고 있다(즉 문헌[LAM92a]). 우선 일정한 종류의 규약들이 여러가지 처리기박자들중에서 동기화를 관리하는데 요구된다. 이 규약은 망오유에 잘 처리되는 묵인한 결점(fault tolerant)과 적대적인 공격에 잘 처리되는 안전한 결점을 둘 다 겸비해야 한다. 다음으로 통신측들중 하나의 박자기구에서 결점으로부터 나타나는 동기화의 림시적손실이 있다면 성공적인 공격의 기회가 발생할것이다. 마지막으로 망지연의 변수와 예언불가능한 속성때문에 배송된 박자가 정확한 동기화를 관리한다고 기대할수 없다. 그

러므로 임의의 일부인기초절차는 충분히 큰 시간창문에 대하여 공격기회를 최소화하는 충분히 작은 망지연을 적응시키는것을 허용해야 한다.

다른 한편 도전-응답방식은 련결 없는 류형의 응용에 적합하지 못하다. 왜냐하면 그것은 련결 없는 처리의 주요특성을 효과적으로 부정하는 임의의 련결 없는 전송전에 핸드쉐이크(handshake)의 간접소비(overhead)를 요구하기때문이다. 이런 응용에 대하여 동기화에서 박자를 유지하도록 하는 매 측에 의한 일정한 종류의 안전한 시간봉사기와 모순이 없는 시도에 대한 믿음성은 가장 좋은 방식일수도 있다(즉 문헌[LAM92b]).

전통암호방식

5.3에서 론의한바와 같이 전통암호화열쇠의 2수준계층을 리용하여 배송된 환경하에서 통신의 기밀성을 담보할수 있다. 일반적으로 이 방략은 신용 있는 열쇠배포쎈터(KDC)를 리용한다. 망에서 매 통신측은 KDC와 함께 주열쇠라고 알려 진 비밀열쇠를 공유한다. KDC는 대화열쇠로 알려 진 두 측사이를 짧은 시간동안에 련결하는데 리용되는 열쇠를 생성하는데 적합하며 배송을 보호하는 주열쇠를 리용하여 이런 열쇠들을 배송하는데 적합하다. 이 방식은 매우 공통적이다. 실례로 11장의 커버로즈(Kerberos)체계를 보자. 여기서의 론의는 커버로즈기구를 리해하는데 관계된다.

그림 5-9는 5장에서 언급된바와 같이 니드함(Needham)과 스쿄레더(Schroeder)[NEED78]가 최초에 제안한 인증기능을 포괄하는 KDC를 리용하여 비밀열쇠를 배송하는것을 보여 준다. 이 규약을 다음과 같이 개괄할수 있다(다음과 같은 표식을 리용한다. P가 통보문 M을 Q에게 보내는 통신단계를 P→Q: M으로 표시한다).

1. $A \to KDC$: $ID_A \| ID_B \| N_1$
2. $KDC \to A$: $E_{K_a}[K_S \| ID_B \| N_1 \| E_{K_b}[K_S \| ID_A]]$
3. $A \to B$: $E_{K_b}[K_S \| ID_A]$
4. $B \to A$: $E_{K_S}[N_2]$
5. $A \to B$: $E_{K_S}[f(N_2)]$

비밀열쇠 K_a와 K_b를 A와 KDC, B와 KDC사이에 각각 공유한다. 규약의 목적은 대화열쇠를 안전하게 A와 B에게 배송하는것이다. A는 안전하게 단계 2에서 새로운 대화열쇠를 얻는다. 단계 3에서 통보문을 복호화하므로 오직 B만이 리해할수 있다. 단계 4는 K_S에 대한 B의 지식을 반영하고 단계 5는 K_S에 대한 A의 지식을 B에게 담보하며 이것은 한번쓰기정보 N_2의 리용때문에 참신한 통보문임을 B에게 담보한다. 5장에서의 론의로부터 단계 4와 5의 목적은 일정한 류형의 재시동공격을 방지하는것이라는것을 상기하자. 특히 적이 단계 3에서 통보문을 채취할수 있고 그것을 재시동한다면 이것은 일정한 방식으로 B에서 조작들을 분렬시킬것이다.

단계 4와 5의 핸드쉐이크에도 불구하고 규약은 아직 재연형태의 공격에 약하다. 적 X가 낡은 대화열쇠를 손상시킬수 있다고 가정하자. 분명히 이것은 적이 단순히 단계 3을 관찰하고 기록하는것보다 더 적합하지 못한 출현이다. 그럼에도 불구하고 이것은 잠재적인 보안위험이다. X는 A인체하면서 단순히 단계 3을 재연함으로써 낡은 열쇠를 리용하여 B를 속일수 있다. 만일 B가 A와 함께 리용된 선행한 모든 대화열쇠들을 수시로 기억하지 않는다면 B는 이것이 재시동임을 결정할수 없다. C가 단계 4에서 핸드쉐이크통보문을 가로챌수 있다면 C는 단계 5에서 A의 응답인체 할수 있다. 이 점으로부터 C는 가짜통보문

을 B에게 보내고 인증된 대화열쇠를 리용하여 A로부터 오는것처럼 B에게 나타난다.

덴닝(Denning)[DENN81, DENN82]은 니드함/스코레더의 규약에 변경을 가하여 이 약점을 극복하는것을 제기하였는데 그것은 단계 2와 3에서 일부인의 첨가를 포함한다. 그의 제의는 주열쇠 K_a와 K_b는 안전하다는것을 가정하고 다음과 같은 단계들로 이루어진다.

1. A→KDC: $ID_A \| ID_B$
2. KDC→A: $E_{K_a}[K_S \| ID_B \| T \| E_{K_b}[ID_A \| T]]$
3. A→B: $E_{K_b}[K_S \| ID_A \| T]$
4. B→A: $E_{K_S}[N_1]$
5. A→B: $E_{K_S}[f(N_1)]$

T는 A와 B에게 대화열쇠를 오직 생성하기만 하였음을 담보하는 일부인이다. 이리하여 A와 B는 둘 다 열쇠배포가 참신한 교환임을 안다. A와 B는

$$|Clock-T| < \Delta t_1 + \Delta t_2$$

임을 검열함으로써 시기적절성을 검증할수 있다. 여기서 Δt_1는(A 또는 B에서) KDC의 박자와 국부박자사이의 평가된 표준편차이며 Δt_2는 기대되는 망지연시간이다. 매 정점(마디점)이 일정한 표준참조원천지에 대한 자기의 박자를 설정할수 있다. 일부인 T는 안전한 주열쇠를 리용하여 암호화되므로 적은 낡은 대화열쇠의 지식을 가지고 있다고 할지라도 단계 3의 재시동이 B에 의해 때아닌 때에 검출될수 있으므로 계승할수 없다.

마지막주의점: 단계 4와 5는 시초의 문헌[DENN81]에 포함되지 않았으나 후에 첨가되였다(문헌[DENN82]). 이런 단계들은 B에서 대화열쇠의 수신을 확인한다.

덴닝규약은 니드함/스코레더규약에 비하여 보안의 높은 정도를 제공한것으로 보인다. 그러나 새로운것이 발생한다. 다시말하여 이 새로운 방식은 망을 통하여 동기화되는 박자에 대한 신뢰를 요구한다. 문헌[GONG92]에서는 위험이 있다는것을 지적하였다. 이 위험은 배송된 박자가 박자결점 또는 동기화기구에 대한 방해의 결과에 의해 동기화되지 않을수 있다는 사실에 기초하고 있다(이런 현상이 생길수 있다. 최근년간에 불량인 소편들이 일련의 콤퓨터와 다른 전자체계들에서 리용되여 날자와 시간을 축적한다. 소편은 1일을 향하여 도약하는 경향을 가진다[NEUM90]).

이 문제는 송신자의 박자가 지향된 수신자박자보다 앞섰을 때 발생한다. 이 경우에 적은 송신자로부터 통보문을 가로챌수 있어서 그것을 후에 통보문의 일부인이 수신자의 싸이트로 될 때 재시동할수 있다. 이 재시동이 바라지 않는 결과를 야기시킬것이다. 공(Gong)은 이런 공격을 억압-재시동공격이라고 하였다.

억압-재시동공격에 대항하는 한가지 방법은 통신측들이 정기적으로 KDC의 박자와 자기의 박자를 검열하는것이다. 박자동기화의 요구를 피하는 다른 또 하나의 방법은 한번쓰기정보를 리용하여 핸드쉐이크규약을 신뢰하는것이다. 이 후자의 방법은 수신자가 미래에 선택할 한번쓰기정보는 송신자를 예견할수 없기때문에 억압-재시동공격에 약하지 않다. 니드함/스코레더규약은 한번쓰기정보를 신뢰하지만 본바와 같이 다른 약점을 가진다.

문헌[KEHN92]에서는 억압-재시동공격에 응답하는 하나의 시도를 만들었으며 같은 시각에 니드함/스코레더규약에서 이 문제를 고정하였다. 계속하여 이 후자의 규약에서

모순성이 강조되였고 개선된 방략이 문헌[NEUM93a]에서 제기되였다(이런것을 정확하게 얻는것은 정말로 힘들다). 규약은 다음과 같다.

1. $A \rightarrow B$: $ID_A \| N_a$
2. $B \rightarrow KDC$: $ID_B \| N_b \| E_{K_b}[ID_A \| N_a \| T_b]$
3. $KDC \rightarrow A$: $E_{K_a}[ID_B \| N_a \| K_S \| T_b] \| E_{K_b}[ID_A \| K_S \| T_b] \| N_b$
4. $A \rightarrow B$: $E_{K_b}[ID_A \| K_S \| T_b] \| E_{K_S}[N_b]$

다음과 같이 단계별로 이 교환을 따르자.

1. A는 한번쓰기정보 N_a를 생성하고 N_a+평문에서의 B에 대한 식별자를 보냄으로써 인증교환을 초기화한다. 이 한번쓰기정보는 A에서 그의 시기적절성을 담보하면 대화열쇠를 포함하는 암호화가 통보문에서 A에게로 귀환될것이다.
2. B는 대화열쇠가 요구된다고 KDC에 경보를 울린다. KDC에서 그의 통보문은 그 식별자와 한번쓰기정보 N_b를 포함한다. 이 한번쓰기정보는 B에게로 귀환될것이다. KDC에서 B의 통보문 역시 B와 KDC에 의하여 공유된 비밀열쇠로 암호화한 블로크를 포함한다. 이 블로크를 리용하여 A에게 증명서를 발행하는 KDC에게 지시한다. 그리고 그 블로크는 지향된 증명서의 수신자,증명서에 제의된 유효시간과 A로부터 수신된 한번쓰기정보를 서술한다.
3. KDC는 B의 한번쓰기정보와 B가 KDC와 공유한 비밀열쇠로 암호화된 블로크를 A에게 통과시킨다. 블로크는 보게 되는바와 같이 그다음의 인증을 위해 A가 리용할수 있는 《표》로서 종사한다. KDC는 또한 A에게 A와 KDC에 의해 공유된 비밀열쇠로 암호화한 블로크를 보낸다. 이 블로크는 B가 A의 초기통보문(ID_B)을 수신하고 이것이 시기적절한 통보문이지만 재시동(N_a)은 아님을 검증하며 이것은 A에게 대화열쇠(K_S)와 그 리용에 관한 시간한계(T_b)를 제공한다.
4. A는 표를 B의 한번쓰기정보, 대화열쇠로 암호화된 후자와 함께 B에게 전송한다. 표는 한번쓰기정보를 얻기 위해 $E_{K_S}[N_b]$를 복호화하는데 리용되는 비밀열쇠를 가진 B를 제공한다. B의 한번쓰기정보가 대화열쇠로 암호화된 사실은 통보문이 A로부터 왔으며 재시동이 아니라는것을 인증한다.

이 규약은 A와 B가 안전한 대화열쇠를 가지고 대화를 확립하는 효과적이고 안전한 수단을 제공한다. 더 나아가서 규약은 반복적으로 인증봉사기와 접속할 필요성을 피하면서 B에게 다음번 인증에 리용될수 있는 열쇠의 소유에서 A를 떼낸다. A와 B가 앞에서 언급한 규약을 리용하여 대화를 확립하고 그다음에 그 대화를 결론짓는다. 그다음에는 규약에 의해 확립된 시간한도내에서 A는 B와 함께 새로운 대화를 요구한다. 다음과 같은 규약이 발생한다.

1. $A \rightarrow B$: $E_{K_b}[ID_A \| K_S \| T_b], N'_a$
2. $B \rightarrow A$: $N'_b, E_{K_S}[N'_a]$
3. $A \rightarrow B$: $E_{K_S}[N'_b]$

B가 단계 1에서 통보문을 수신할 때 표가 유효기간이 넘었는가를 검증한다. 새롭게 생성된 한번쓰기정보 N'_a 와 N'_b 는 재시동공격이 없다는것을 매 부분에게 담보한다.

앞에서 언급한 모든것들에서 T_b로 서술된 시간은 B의 박자에 관계되는 시간이다. 이리하여 이 일부인은 B가 오직 자체의 생성된 일부인만을 검열하므로 박자를 동기화할것을 요구하지 않는다.

공개열쇠암호방식

6장에서는 대화열쇠배포를 위하여 공개열쇠암호를 리용하는 한가지 방식을 제기하였다(그림 6-15). 이 규약은 쌍방중 일방이 다른 일방의 현재공개열쇠를 소유하는것을 가정한다. 이 가정을 요구하는것은 실천적이 못될수도 있다.

일부인을 리용하여 한가지 규약을 문헌[DENN81]에서는 다음과 같이 제공하였다.

1. A→AS: $ID_A \| ID_B$
2. AS→A: $E_{KR_{as}}[ID_A \| KU_a \| T] \| E_{KR_{as}}[ID_B \| KU_b \| T]$
3. A→B: $E_{KR_{as}}[ID_A \| KU_a \| T] \| E_{KR_{as}}[ID_B \| KU_b \| T] \| E_{KU_b}[E_{KR_a}[K_S \| T]]$

이 경우에 중심체계를 인증봉사기(AS)라고 한다. 왜냐하면 비밀열쇠배포를 위해서는 실제적으로 적합하지 못하기때문이다. 오히려 AS는 공개열쇠확인을 담보한다. 대화열쇠를 선택하고 A로 암호화한다. 따라서 AS에 의해 적발될 위험은 없다. 일부인은 손상된 열쇠의 재시동으로부터 보호된다.

이 규약은 조밀하지만 앞에서처럼 박자의 동기화를 요구한다. 우(Woo)와 람(Lam)의 문헌[WOO92a]에 의해 제안된 다른 한가지 방식은 한번쓰기정보를 리용하는것이다. 이 규약은 다음과 같은 단계들로 이루어 진다.

1. A→KDC: $ID_A \| ID_B$
2. KDC→A: $E_{KR_{auth}}[ID_B \| KU_b]$
3. A→B: $E_{KU_b}[N_a \| ID_A]$
4. B→KDC: $ID_B \| ID_A \| E_{KR_{auth}}[N_a]$
5. KDC→B: $E_{KR_{auth}}[ID_A \| KU_a] \| E_{KU_b}[E_{KR_{auth}}[N_a \| K_S \| ID_B]]$
6. B→A: $E_{KU_a}[E_{KR_{auth}}[N_a \| K_S \| ID_B] \| N_b]$
7. A→B: $E_{K_S}[N_b]$

단계 1에서 A는 B와의 안전한 련결을 확립하는데로 지향된 KDC에게 통지한다. KDC는 B의 공개열쇠확인의 복사를 A에게 돌려 준다(단계 2). B의 공개열쇠를 리용하여 A는 B에게 한번쓰기정보 N_a를 통신하여 보낼것을 통지한다(단계 3). 단계 4에서 B는 A의 공개열쇠확인을 위해 KDC에게 부탁하여 대화열쇠를 요구한다. 그리고 B는 A의 한번쓰기정보를 포함하므로 KDC는 그 한번쓰기정보로 대화열쇠를 일부인할수 있다. 한번쓰기정보는 KDC의 공개열쇠를 리용하여 보호된다. 단계 5에서 KDC는 B에게 A의 공개열쇠확인의 복사+정보{N_a, K_S, ID_B}를 돌려 준다. 이 정보는 기본적으로 K_S는 B의 이름으로 KDC에 의해 생성되고 N_a에 관계되는 비밀열쇠이라는것을 의미한다. K_S와

N_a에 대한 속박은 K_S가 참신하다는것을 A에게 담보한다. 이 3항조는 사실상 KDC로부터 온다는것을 검증하기 위하여 B에게 허용하는 KDC의 비밀열쇠를 리용하여 암호화된다. 또한 B의 공개열쇠를 리용하여 암호화되여 다른 그 어떤 실체도 A와 부정적으로 련결을 확립하기 위한 시도에서 3항조를 리용하지 않을수 있다. 단계 6에서 여전히 KDC의 비밀열쇠에 의해 암호화된 3항조 $\{N_a, K_S, ID_B\}$는 B에 의해 생성된 한번쓰기정보 N_b와 함께 A에게 중계된다. 선행한 모든것은 A의 공개열쇠를 리용하여 암호화된다. A는 대화열쇠 K_S 를 받아서 그것을 리용하여 N_b를 암호화하며 그것을 B에게 돌려 준다. 이 마지막통보문은 B에게 대화열쇠에 대한 A의 지식을 담보한다.

이것은 여러가지 공격을 고려한 안전한 규약으로 보인다. 그러나 저자들자신은 결점을 발견하고 문헌[WOO92b]에서 다음과 같은 알고리듬의 개정판을 제출하였다.

1. A→KDC: $ID_A \| ID_B$
2. KDC→A: $E_{KR_{auth}}[ID_B \| KU_b]$
3. A→B: $E_{KU_b}[N_a \| ID_A]$
4. B→KDC: $ID_B \| ID_A \| E_{KU_{auth}}[N_a]$
5. KDC→B: $E_{KR_{auth}}[ID_A \| KU_a] \| E_{KU_b}[E_{KR_{auth}}[N_a \| K_S \| ID_A \| ID_B]]$
6. B→A: $E_{KU_a}[E_{KR_{auth}}[N_a \| K_S \| ID_A \| ID_B] \| N_b]$
7. A→B: $E_{K_S}[N_b]$

A의 식별자 ID_A는 단계 5와 6에서의 KDC의 비밀열쇠로 암호화한 항목들의 모임에 첨가된다. 이것은 대화열쇠 K_S를 대화에 의해 련결될 두 부분의 신원에 속박시킨다. 이 ID_A의 포함은 한번쓰기정보값 N_a가 A에 의해 생성된 모든 한번쓰기정보들에서만 오직 유일하게 고찰되나 모든 통신측들에 의해 생성된 한번쓰기정보모두에서는 그렇지 않다는 사실을 보여주고 있다. 이리하여 그것은 유일하게 A의 련결요구를 동일시하는 쌍$\{ID_A, N_a\}$이다.

이 실례와 앞에서 서술된 규약에서 둘 다 안전한것 같은 규약은 보충적인 해석후에 개정되였다. 이런 실례들은 인증분야에서 첫 시각에 아주 좋은것을 설정하는것은 힘들다는것을 강조하고 있다.

한방향인증

암호가 급진적으로 성장하는 한가지 응용은 전자우편이다. 전자우편의 속성과 그의 주요우점은 송수신자가 같은 시각에 직결적으로 하는것이 필요없다는것이다. 그대신에 전자우편통보문은 수신자의 전자우편통에로 향하는데 그것은 수신자가 그것을 읽어 낼때까지 완충된다.

전자우편통보문의 《봉투》나 머리부는 분명한 곳에 있어야 하며 따라서 통보문을 단순우편전송규약(SMTP)이나 X.400과 같은 《기억-그리고-향하기》전자우편규약에 의해 조종할수 있다. 그러나 때때로 우편조종규약은 통보문의 평문형태를 참조하지 않을것을 바란다. 왜냐하면 우편조종기구를 신뢰할것을 요구하기때문이다. 따라서 전자우편통보문은 암호화되여 우편조종체계가 복호화열쇠를 소유하지 않는다. 다음요구는 인증이다. 전형적으로 수신자는 통보문이 제정된 송신자로부터 왔다는것을 담보하기 바란다.

전통암호방식

전통암호를 리용하여 그림 5-11에서 보여 준 분산된 열쇠배포씨나리오는 실천적이지 못된다. 이 방식은 송신자가 지향된 수신자에 대한 요청을 발행할것을 요구하고 대화열쇠를 포함하는 응답을 기다리며 오직 그때에만 통보문을 보낸다.

일련의 세분들에 대하여 그림 5-9로 보여 준 KDC방략이 암호화된 전자우편에 대한 후보자이다. 수신자(B)가 같은 시각에 송신자(A)와 직결적으로 있을 요구를 피하려고 하므로 단계 4와 5는 소거되여야 한다. 내용 M인 통보문에 대하여 렬은 다음과 같다.

1. A→KDC: $ID_A \| ID_B \| N_1$
2. KDC→A: $E_{K_a}[K_S \| ID_B \| N_1 \| E_{K_b}[K_S \| ID_A]]$
3. A→B: $E_{K_b}[K_S \| ID_A] \| E_{K_S}[M]$

이 방식은 오직 통보문의 지향된 수신자만이 그것을 읽을수 있다는것을 담보한다. 이것은 또한 송신자가 A인 인증의 준위를 제공한다. 설명한바와 같이 규약은 재시동으로부터 보호되지 못한다. 통보문과 일부인을 포함함으로써 몇가지 방어수단이 제공된다. 그러나 전자우편에서 잠재적지연때문에 이런 일부인은 유용성에 한계가 있을것이다.

공개열쇠암호방식

이미 기밀성(그림 8-1의 ㄴ), 인증(그림 8-1의 ㄷ) 또는 둘 다(그림 8-1의 ㄹ)에 대한 전체적인 통보문의 간단한 암호화를 포함하는 전자우편에 적합한 공개열쇠암호방식을 제기하였다. 이 방식은 송신자가 수신자의 공개열쇠(기밀성)를 알거나 수신자가 송신자의 공개열쇠(인증)또는 둘 다(기밀성+인증)를 알것을 요구한다. 그외에도 공개열쇠알고리듬은 긴 통보문에 한번 또는 두번 적용되여야 한다.

기밀성이 기본이라면 다음과 같은것이 더 효과적일것이다.

$$A \rightarrow B:\ E_{KU_b}[K_S] \| E_{K_S}[M]$$

이 경우에 통보문은 한시각(one-time)비밀열쇠로 암호화된다. A 역시 이 한시각열쇠를 B의 공개열쇠로 암호화한다. 오직 B만이 대응하는 비밀열쇠를 리용하여 한시각열쇠를 얻을수 있으며 그다음에는 그 열쇠를 리용하여 통보문을 복호화할수 있다. 이 방식은 B의 공개열쇠로 전체적인 통보문을 간단히 암호화하는것보다 더 효과적이다.

인증이 기본이라면 수자서명은 그림 8-5에서 서술한바와 같이 다음과 같은것을 만족시킬수 있다.

$$A \rightarrow B:\ M \| E_{KR_a}[H(M)]$$

이 방법은 A가 후에 통보문을 보낸 사실을 부인할수 없음을 담보한다. 그러나 이 기술은 다른 한가지 종류의 사기행위를 공개적으로 하게 한다. 보브(Bob)는 회사돈을 보관하려고 하는 자기의 책임자 알리스(Alice)에게 보낼 통보문을 구성한다. 그는 자기의 서명을 보증하며 그것을 전자우편체계에 보낸다. 결국 통보문은 알리스의 우편통에 전달될것이다. 그러나 막스(Max)가 보브의 의도를 알고 전달하기전에 우편렬들에 대한 참조를 얻는다고 가정하자. 그는 보브의 통보문을 구하여 그의 서명을 벗기여 내고 그를

보증하여 알리스에게 전달될 통보문을 재정돈한다. 막스는 보브의 의도에 대한 신용을 얻는다.

이런 방식에 대항하기 위하여 통보문과 서명을 둘 다 수신자의 공개열쇠로 다음과 같이 암호화할수 있다.

$$A \rightarrow B: E_{KU_b}[M \| E_{KR_a}[H(M)]]$$

마지막 두 방식은 B가 A의 공개열쇠를 알고 그것이 시기적절함을 확신할것을 요구한다. 이 담보를 제공하는 효과적인 하나의 방법은 6장에서 론의한 수자확인이다. 이제는

$$A \rightarrow B: M \| E_{KR_a}[H(M)] \| E_{KR_{as}}[T \| ID_A \| KU_a]$$

이다.

통보문외에도 A는 B에게 A의 비밀열쇠로 암호화된 서명과 인증봉사기의 비밀열쇠로 암호화된 A의 확인을 보낸다. 통보문의 수신자는 우선 확인을 리용하여 송신자의 공개열쇠를 얻어서 그것이 인증된다는것을 검증한 다음에는 공개열쇠를 리용하여 통보문자체를 검증한다. 기밀성이 요구되면 전체적인 통보문을 B의 공개열쇠로 암호화할수 있다. 다른 한가지 방법으로서 전체적인 통보문을 한시각비밀열쇠로 암호화할수 있다. 그리고 비밀열쇠 역시 전송되고 B의 공개열쇠로 암호화된다. 이 방식을 12장에서 설명한다.

10.3 수자서명표준

국가표준 및 기술연구소(NIST)는 수자서명표준(DSS)으로 알려 진 련방정보처리표준 FIPS PUB 186을 발표하였다. DSS는 9장에서 묘사한 안전한 하쉬알고리듬(SHA)을 리용하여 새로운 수자서명기술인 수자서명알고리듬(DSA)을 제기하였다. DSS는 원래 1991년에 제기되였으며 방식의 보안에 관련되는 공개적인 반결합에 응답하여 1993년에 수정되였다. 그후 1996년에 좀 더 개정되였다.

DSS 방식

DSS는 수자서명함수만을 제공하도록 설계된 알고리듬을 리용한다. RSA와는 달리 이것은 암호화나 열쇠교환에 리용될수 없다. 그럼에도 불구하고 이것은 공개열쇠기술이다.

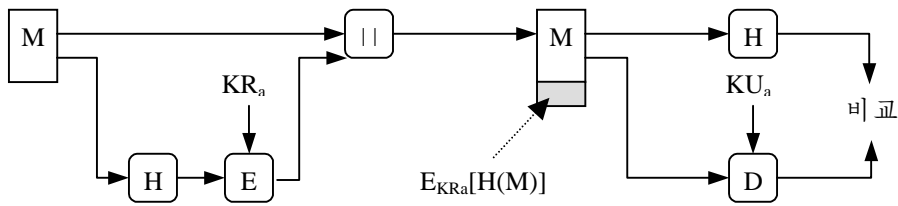
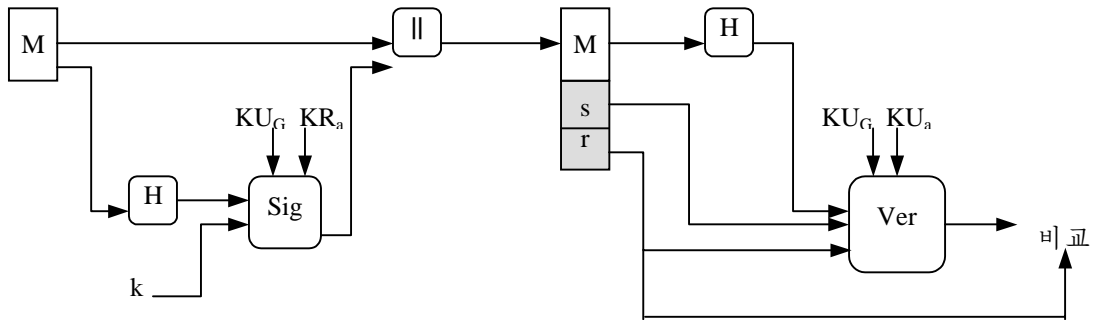
그림 10-1은 수자서명을 생성하기 위한 DSS방식과 RSA를 리용한 방식을 대비한다. RSA방식에서는 서명되는 통보문이 하쉬함수에 입력되여 고정길이의 안전한 하쉬부호를 생성해 낸다. 그다음 이 하쉬부호는 서명을 형성하는 송신자의 비밀열쇠를 리용하여 암호화된다. 다음에 통보문과 서명은 둘 다 전송된다. 수신자는 통보문을 취하여 하쉬부호를 생성해 낸다. 수신자는 또한 송신자의 공개열쇠를 리용하여 서명을 복호화한다. 계산된 하쉬부호가 복호화된 서명과 정합되면 서명을 타당한것으로 접수한다. 오직 송신자만이 비밀열쇠를 알고 있으므로 송신자만이 타당한 서명을 생성해 낼수 있을것이다.

ㄱ) RSA 방식

ㄴ) DSS 방식

그림 10-1. 수자서명에 대한 두가지 방식

DSS방식도 역시 하쉬함수를 리용한다. 하쉬부호는 이 특별한 서명에 대하여 생성된 우연수 k와 함께 서명함수에로의 입력으로서 주어 진다. 서명함수 역시 송신자의 비밀열쇠(KR_a)와 통신원리들의 그룹인 파라메터들의 모임에 의거한다. 하나의 대역적공개열쇠(KU_G)를 구성하는 이 모임을 고찰할수 있다(매 사용자에 따라 변하는 이런 보충적인 파라메터들을 허용하는것 역시 가능하며 그래서 그것들은 사용자의 공개열쇠의 한부분이다. 실천에서는 대역적공개열쇠를 매 사용자의 공개열쇠와 구별하는데 리용하는것이 더 적합하다). 결과는 s와 r로 표시한 두개 성분들로 구성된 서명이다.

수신하는 말단에서 들어 오는 통보문의 하쉬부호를 생성한다. 이것+서명이 검증함수에로의 입구로 된다. 검증함수 역시 송신자의 공개열쇠(KU_a)는 물론 대역적공개열쇠에 의거하는데 이것은 송신자비밀열쇠와 함께 쌍을 이룬다. 검증함수의 출력은 서명이 타당하면 서명성분 r와 같은 값이다. 서명함수는 오직 비밀열쇠의 지식을 가진 송신자들만이 타당한 서명을 생성해 내는 함수이다.

이제 이 알고리듬을 구체적으로 보자.

수자서명알고리듬

DSA는 리산로그계산의 어려움성에 기초하고 있으며(7장을 보시오) 엘가말의 문헌[ELGA85]과 스츠노(Schnorr)의 문헌[SCHN91]에 의해 최초에 제안된 방식에 기초하고 있다.

대역적공개열쇠성분
p 512≤L≤1024이고 L은 64의 배수 즉 64bit의 증분으로서 512와 1024bit사이의 비트길이이며 $2^{L-1}<p<2^L$인 씨수
q $2^{159}<q<2^{160}$ 즉 160bit의 비트길이인 (p-1)의 씨약수
g = $h^{(p-1)/q}$ mod p. 여기서 h는 1<h<(p-1)이고 $h^{(p-1)/q}$ mod p>1인 임의의 옹근수이다.

사용자의 비밀열쇠
x 0<x<q인 우연 또는 준우연옹근수

사용자의 공개열쇠
y = g^x mod p

사용자의 단위당 통보문비밀수
k = 0<k<q인 우연 또는 준우연옹근수

서 명
r = (g^x mod p) mod q
s = [k^{-1}(H(M)+xr)] mod q
서명 = (r, s)

검 증
w = $(s')^{-1}$mod q
u_1 = [H(M')w] mod q
u_2 = (r')w mod q
v = [($g^{u1}y^{u2}$) mod p] mod q
TEST: v=r'

M : 서명할 통보문
H(M) : SHA-1을 리용한 M의 하쉬
M', r', s': M, r, s의 수신판

그림 10-2. 수자서명알고리듬(DSA)

그림 10-2는 이 알고리듬을 개괄한다. 공개이고 사용자집단이 공통으로 가지는 3개의 파라메터들이 있다. 16bit씨수 q를 선택한다. 다음으로 씨수 p를 512bit와 1024bit사이길이로 q가 (p-1)을 나누도록 선택한다. 최종적으로 g가 $h^{(p-1)/q}$ mod p형태를 가지도록 선택한다. 여기서 h는 g가 1보다 커야 한다는 제한을 가진 1과 (p-1)사이의 옹근수이다(수론적용어로 g는 p에 관한 q의 차수이다. 7장을 보시오).

이런 수들을 가지고 매 사용자들은 비밀열쇠를 선택하고 공개열쇠를 생성한다. 비밀열쇠 x는 1부터 $(q-1)$사이의 수이여야 하며 우연적으로 또는 준우연적으로 선택될것이다. 공개열쇠를 비밀열쇠로부터 $y=g^x \bmod p$로 계산한다. x가 주어 졌을 때 y의 계산은 상대적으로 간단하다. 그러나 공개열쇠 y가 주어 졌을 때 mod P에 관하여 밑수 g에 대한의 리산로그인 x를 결정하는것이 계산량적으로 불가능할것이라고 믿는다(7장을 보시오).

서명을 창조하기 위하여 사용자는 2개의 량 r와 s를 계산하는데 이것들은 공개열쇠성분 (p, q, g), 사용자의 비밀열쇠(x), 통보문의 하쉬부호 H(M) 그리고 보충적인 옹근수 k에 관한 함수로서 k는 우연 또는 준우연적으로 생성되며 매 서명에 대해 유일하다.

수신하는 말단에서 검증은 그림 10-2에서 보여 준 공식을 리용하여 진행된다. 수신자는 공개열쇠성분, 송신자의 공개열쇠 그리고 들어 오는 통보문의 하쉬부호에 관한 함수인 량 v를 생성한다. 이 량이 서명의 성분 r와 정합되면 서명은 타당하다.

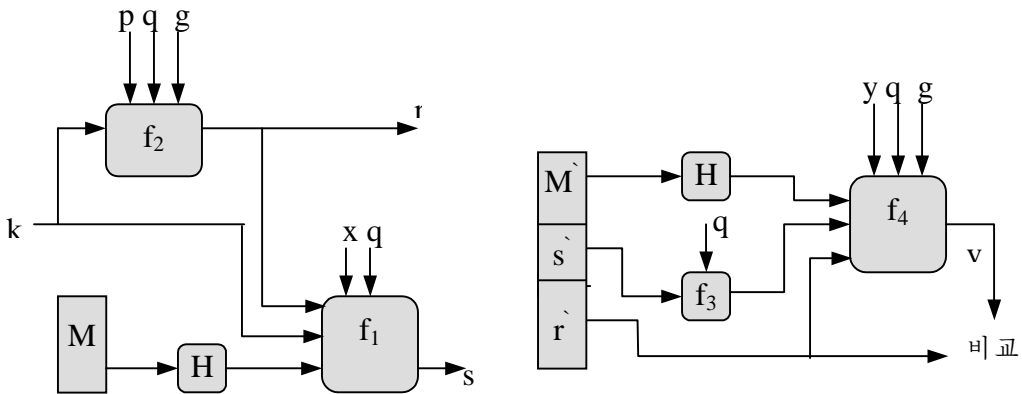



그림 10-3. DSS서명과 검증

그림 10-3은 서명과 검증에 관한 함수를 묘사한다.

그림 10-3에서 보여 준바와 같이 알고리듬의 구조는 아주 흥미 있다. 말단에서의 검사는 값 r에 있다는것을 강조하게 되는데 이것은 통보문에 전혀 관계되지 않는다. 그대신에 r는 k와 3개의 대역적공개열쇠성분에 관한 함수이다. k의 (mod P에 관한)곱하기역수를 통보문하쉬부호와 사용자의 비밀열쇠를 입력으로 하는 함수에로 통과시킨다. 이 함수의 구조는 수신자가 들어 온 통보문과 서명, 사용자의 공개열쇠 그리고 대역적공개열쇠를 리용하여 r를 구할수 있도록 한것이다. 그림 10-2나 10-3으로부터 이런 방식으로 동작하는것은 확실히 명백하지 못하다. 증명은 부록 10에 준다.

리산로그를 취하는 어려움성으로 하여 적은 r로부터 k를 구하거나 s로부터 x를 구하는것이 불가능하다.

다른 한가지 강조할것은 서명생성에서는 오직 계산량적으로 요구되는 과제가 mod P에 관한 제곱연산 g^k이다. 이 값은 서명되는 통보문에 관계되지 않으므로 미리 계산해놓을수 있다. 사실 사용자는 필요에 따라 문서를 서명하는데 리용되는 일련의 r의 값들을 미리 계산해 놓는다. 오직 차이나는것은 곱하기역수 k^{-1}을 결정하는것이다. 다시 일련의 이 값들을 미리 계산할수 있다.

참고문헌

문헌[AKL83]은 수자서명에 관한 고전문헌으로서 아직 많이 리용되고 있다. 최근에는 문헌[MITC92]가 좋을것이다.

AKL83 Akl, S. "Digital Signatures: A Tutorial Survey." *Computer*, February 1983.

MITC92 Mitchell, C.; Piper, F.; and Wild, P. " Digital Signatures. " In [SIMM92a].

문 제

1. 10.2에서는 비밀열쇠의 배송을 위하여 문헌[WOO92a]에서 제기한 공개열쇠방식을 개괄하였다. 개정판은 단계 5와 6에서 ID_A를 포함한다. 이 개정판에 의하여 특별히 어떤 공격에 대항할수 있는가?
2. 문제 10-1로 귀착된 규약의 7개 단계로부터 다음과 같은 렬들을 가지는 5개 단계로 줄일수 있다.
   1. A→B:
   2. B→KDC:
   3. KDC→B:
   4. B→A:
   5. A→B

   매 단계에서 전송된 통보문을 보여 주시오. 암시: 이 규약에서 최종통보문은 원래 규약에서의 최종통보문과 같다.
3. 표 10-1의 ㄱ과 10-1의 ㄴ의 수자서명기술을 변경하여 수신자가 서명을 검증할수 있게 하시오.
4. 표 10-1의 ㄷ의 수자서명기술을 변경하여 완전한 통보문의 3중암호화를 피하시오.
5. 표 10-1의 ㄹ을 론의하는데서 속임수는 불가능하다는것을 규정하였다. 사실 하나의 가능성이 있다. 그것을 서술하고 안전하게 무시할수 있는 그런 작은 신뢰성이 어떻게 나오는가를 설명하시오.
6. 10.2에서 서술한 억압-재시동공격을 참고하여

   ㄱ) 통신측의 박자가 KDC의 박자우에 있을 때의 공격실례를 드시오.

   ㄴ) 통신측의 박자가 다른 한 측의 박자우에 있을 때의 공격실례를 드시오.

7. 도전으로서 한번쓰기정보를 리용하는 3개의 전형적인 방법들이 있다. N_a는 A에 의해 생성된 한번쓰기정보이고 A와 B는 열쇠 K를 공유하며 f( )는 이런 증가함수이다. 3개의 사용법은 다음과 같다.

사용법1	사용법2	사용법3
(1) $A \to B$: N_a (2) $A \to B$:$E_K[N_a]$	(1) $A \to B$:E_K $[N_a]$ (2) $A \to B$: N_a	(1) $A \to B$:$E_K[N_a]$ (2) $A \to B$:$E_K[f(N_a)]$

매 사용법의 적당한 경우를 묘사하시오.

8. 와트손(Watson)은 홈스(Holmes)의 일이 끝나기를 참을성 있게 기다렸다. 《홈스, 몇가지 흥미 있는 문제들을 풀수 있나?》 그는 홈스가 일을 다 끝냈을 때 물었다. 《아, 아닐세. 일상적으로 하던 화학실험대신에 나의 전자우편을 검사하고 그다음에 한두가지 망실험을 했을뿐이야. 나에게는 지금 의뢰자가 한명밖에 없는데 난 벌써 그의 문제를 풀었거든. 나의 기억이 틀리지 않는다면 자네가 언젠가 한번 암호기술도 자네의 취미중의 하나라고 말했던것같은데. 그렇다면 그것이 아마 자네의 흥미를 끌수 있어.》

《그렇기는 한데. 홈스, 난 다만 암호학에 대한 애호가일따름이야. 그러나 물론 나는 그 문제에 흥미가 있어. 그게 대체로 어떤거요?》

《내 의뢰인은 호스그래브(Hosgrave)라는 사람인데 비교적 자그마하고 경기가 좋은 은행의 리사일세. 그 은행은 완전히 콤퓨터화되여서 망통신을 광범히 사용하지. 그 은행은 이미 RSA를 리용하여 자기의 자료를 보호하고 전달하는 문서들을 수자식으로 서명하고 있네. 그 은행은 지금 자기의 절차들에서 몇가지 전환을 하려고 하네. 특히 일부 문서들을 다음과 같이 두명의 서명자에 의하여 수자식으로 서명하려고 하네.》

ㄱ) 첫번째 서명자는 문서를 준비하고 서명한 다음에 그것을 두번째 서명자에게 보낸다.

ㄴ) 두번째 서명자는 첫 단계로 문서에 첫번째 서명자가 진짜서명하였는가를 확인한다. 그다음 수신자뿐아니라 집단의 그 어느 사람도 두 사람이 다 서명하였다는것을 확인할수 있도록 문서에 자기서명을 한다. 또한 두번째 서명자만이 단계 1이 끝난 후에 문서에 대한 서명을 확인할수 있어야 한다. 즉 수신자(또는 집단의 임의의 성원)가 한사람의 서명만 있는 중간단계의 문서가 아니라 두 사람의 서명이 다 있는 완성된 문서만을 확인할수 있게 되여야 한다. 또한 그 은행은 RSA수자식서명을 지원하는 현존 모듈을 리용하고 싶어한다. 》

《음. 홈스, RSA를 리용하여 어떻게 한사람이 수자식문서를 서명할수 있는가가 리해되는구만. 자네가 RSA수자식서명의 적당한 일반화를 리용하여 호스그래브의 문제를 풀었다고 난 추측하는데.》

《맞았어, 와트손. 》 샬록크 홈스가 머리를 끄떡이였다. 《원래 RSA수자식서명은 서명자의 비밀복호화열쇠 <d>를 가지고 문서를 암호화하는 방법으로 이루어 졌는데 서명은 널리 알려 진 암호열쇠 <e>를 리용하여 그 복호화를 통하여 임의의 사람이 확인할수 있었네. 누구나 서명 S가 d를 알고 있는 사람에 의하여 진행되였다고 확인할수 있는데 그것은 유일한 서명자라고 추측되지. 지금 호스그래브의 문제는 그 과정을 약간 일반화함으로써 같은 방법으로 풀수 있네.》

이상의 내용에서 호스그래브의 문제를 푸시오.

9. DSA는 서명-생성처리가 s=0의 값으로 나타나면 k의 새로운 값이 생성되여 서명이 다시 계산되도록 묘사한다. 왜 그런가?
10. DSA를 창조하는데 리용된 k값이 손상되면 어떤 현상이 일어나는가?
11. DSS문서는 다음과 같은 씨수판정알고리듬을 포함한다.

(1) [***w*의 선택**] w가 우연홀수라고 하자. 이때 (w-1)은 짝수로서 m이 홀수인 $2^a m$형태로 표시될수 있다. 즉 2^a은 (w-1)을 나누는 2의 최대제곱이다.
(2) [***b*의 생성**] b를 $1<b<w$인 우연옹근수라고 하자.
(3) [**지수계산**] j=0이고 $z=b^m \bmod w$로 설정한다.
(4) [**다 했는가?**] j=0이고 z=1이거나 $z=w-1$이면 w는 검사를 통과하여 씨수일수 있으며 단계 8에로 간다.
(5) [**완료하는가?**] $j>0$이고 z=1이면 w는 씨수아니며 이 w에 대한 알고리듬을 완료한다.
(6) [***j*의 증가**] $j=j+1$로 놓는다. $j<a$이면 $z=z^2 \bmod w$으로 놓고 단계 4에로 간다.
(7) [**완료**] w가 씨수아니고 이 w에 대한 알고리듬을 완료한다.
(8) [**다시 검사하겠는가?**] b의 충분한 우연값들이 검사되였다면 w를 씨수로 접수하고 알고리듬을 완료하며 그렇지 않으면 단계 2에로 간다.

ㄱ) 알고리듬이 어떻게 동작하는가를 설명하시오.
ㄴ) 7장에서 론의된 밀러-라빈검사와 동등하다는것을 보여 주시오.

12. DSS에 대하여 k의 값은 매 서명에 대하여 생성되므로 같은 통보문이 서로 다른 기회에 두번 서명된다고 하더라도 서명은 서로 다르다. 이것은 RSA서명에 대해서는 참이 아니다. 이 차이의 실천적인 따름은 무엇인가?
13. 수자서명으로 리용되는 디피-헬만에 대한 변종을 개발하려고 시도하고 있다. 여기에는 DSA보다 단순하며 다음과 같이 비밀열쇠에 보충적으로 비밀함수를 요구하지 않는것이 있다.

공개요소:
q 씨수
α $\alpha < q$이고 α는 q의 원시뿌리
비밀열쇠:
X $X < q$
공개열쇠:
$Y= \alpha^X \bmod q$

통보문 M을 서명하기 위하여 통보문의 하쉬부호인 h=H(M)을 계산한다. gcd(h, p-1)=1일것을 요구한다. 그렇지 않다면 통보문에 하쉬를 첨가하고 새로운 하쉬를 계산한다. 하쉬부호가 (p-1)과 서로 소인것을 생성해 낼 때까지의 처리를 계속한다. 그다음에 $Z \times h = X \bmod (q-1)$를 만족시키는 Z를 계산한다. 통보문의 서명은 α^Z이다. 서명을 검증하기 위하여 사용자는 $(\alpha^Z)^h = \alpha^X \bmod q$임을 검증한다.

ㄱ)이 방식의 동작과정을 보여 주시오. 즉 증명처리가 서명이 타당하면 등식을 생성해 낸다는것을 보여 주시오.
ㄴ)임의의 통보문에 대한 사용자의 서명을 위조하는 단순한 기술을 묘사함으로써 이 방식이 접수불가능함을 보여 주시오.

부록 10. DSS알고리듬의 증명

이 부록의 목적은 서명검증에서 서명이 타당하면 $v = r$이라는 증명을 제공하는것이다. 다음과 같은 증명은 FIPS표준에서 나타나지만 유도담당자가 해야 할 보충적이고 구체적인것을 포함하는것에 기초하고 있다.

보조정리1. 임의의 옹근수 t에 대하여

if	$g=h^{(p-1)/q} \bmod p$
then	$g^t \bmod p=g^{t \bmod q} \bmod p$

증명: 페르마의 정리(7장)에 의하여 h는 p와 서로 소이므로 $h^{p-1} \bmod p$=1이다. 따라서 임의의 부아닌 옹근수 n에 대하여

$$\begin{aligned} g^{nq} \bmod p &= (h^{(p-1)/q} \bmod p)^{nq} \bmod p \\ &= h^{((p-1)/q)nq} \bmod p && \text{모드산수의 규칙에 의하여} \\ &= h^{(p-1)n} \bmod p \\ &= ((h^{(p-1)} \bmod p)^n \bmod p && \text{모드산수의 규칙에 의하여} \end{aligned}$$

이리하여 부아닌 옹근수 n과 z에 대하여

$$\begin{aligned} g^{nq+z} \bmod p &= (g^{nq}g^z) \bmod p \\ &= ((g^{nq} \bmod p)(\ g^z \bmod p)) \bmod p \\ &= g^z \bmod p \end{aligned}$$

임의의 부아닌 옹근수 t를 $t = nq + z$로 유일하게 표시할수 있다. 여기서 n과 z는 부아닌 옹근수이며 $0 < n < z$이다. 그래서 $z = t \bmod q$이다. 따라서 결과가 나온다. **증명끝.**

보조정리 2. 임의의 부아닌 옹근수 a와 b에 대하여

$$g^{(a \bmod q+b \bmod q)} \bmod p = g^{(a+b) \bmod q} \bmod p$$

증명: 보조정리 1에 의하여

$$\begin{aligned} g^{(a \bmod q+b \bmod q)} \bmod p &= g^{(a \bmod q+b \bmod q) \bmod q} \bmod p \\ &= g^{(a+b) \bmod q} \bmod p \end{aligned}$$

증명끝.

보조정리3.

$$y^{(rw) \bmod q} \bmod p = g^{(xrw) \bmod q} \bmod p$$

증명: 정의(그림 10-2)에 의하여 $y = g^x \bmod p$이다. 이때

$$
\begin{aligned}
y^{(rw) \bmod q} \bmod p &= (g^x \bmod p)^{(rw) \bmod q} \\
&= g^{x((rw) \bmod q)} \bmod p && \text{모드산수규칙에 의해} \\
&= g^{(x((rw) \bmod q)) \bmod q} \bmod p && \text{보조정리 1에 의해} \\
&= g^{(xrw) \bmod q} \bmod p
\end{aligned}
$$

증명끝.

보조정리4.

$$((H(M)+xr)w) \bmod q = k$$

증명: 정의(그림 10-2)에 의하여 $s=(k^{-1}(H(M)+xr)) \bmod q$이다. 또한 q는 씨수이므로 q보다 작은 임의의 부아닌 옹근수는 곱하기역수를 가진다(7장). 그러므로 $(k\,k^{-1}) \bmod q=1$이다. 이때

$$
\begin{aligned}
(ks) \bmod q &= (k((k^{-1}(H(M)+xr)) \bmod q)) \bmod q \\
&= ((k(k^{-1}(H(M)+xr)))) \bmod q \\
&= (((kk^{-1}) \bmod q)((H(M)+xr) \bmod q)) \bmod q \\
&= ((H(M)+xr)) \bmod q
\end{aligned}
$$

정의에 의하여 $w = s^{-1} \bmod q$이고 따라서 $(ws) \bmod q = 1$이다. 그러므로

$$
\begin{aligned}
((H(M)+xr)w) \bmod q &= (((H(M)+xr) \bmod q)(w \bmod q)) \bmod q \\
&= (((ks) \bmod\ )(w \bmod q)) \bmod q \\
&= (kws) \bmod q \\
&= ((k \bmod q)((ws) \bmod q)) \bmod q \\
&= k \bmod q
\end{aligned}
$$

이다. $0 < k < q$이므로 $k \bmod q = k$이다. **증명끝.**

정리: 그림 10-2의 정의를 리용하여 $v=r$이다.

$$
\begin{aligned}
v &= ((g^{u1}y^{u2}) \bmod p) \bmod q && \text{정의에 의하여} \\
&= (g^{(H(M)w) \bmod q}\, y^{(rw) \bmod q}) \bmod p) \bmod q \\
&= (g^{(H(M)w) \bmod q}\, y^{(xrw) \bmod q}) \bmod p) \bmod q && \text{보조정리 3에 의하여} \\
&= (g^{(H(M)w) \bmod q+(xrw) \bmod q}) \bmod p) \bmod q \\
&= (g^{(H(M)w+xrw) \bmod q}) \bmod p) \bmod q && \text{보조정리 2에 의하여} \\
&= (g^{((H(M)w+xr)w) \bmod q}) \bmod p) \bmod q \\
&= (g^k \bmod p) \bmod q && \text{보조정리 4에 의하여} \\
&= r && \text{정의에 의하여}
\end{aligned}
$$

증명끝.

제3편. 망보안실천

제11장. 인증응용

이 장에서는 응용준위의 인증 및 수자서명을 지원하기 위하여 개발된 일부 인증기능들에 대하여 소개한다.

먼저 이전부터 널리 쓰이고 있는 봉사의 하나인 Kerberos봉사에 대하여 고찰하고 다음에 X.509등록부인증봉사를 고찰한다. 이 표준은 12장에서 론의하는 S/MIME와 같은 다른 표준들을 지원할뿐아니라 가장 기초적인 등록부봉사로서 중요하다.

11.1 KERBEROS

Kerberos는 MIT의 대상과제아데나(Project Athena)의 부분으로 개발된 인증봉사이다. Kerberos는 다음과 같은 문제를 취급한다. 우선 사용자들이 망에 분산된 봉사기들의 봉사에 접근하려고 하는 열린 분산환경을 생각하자. 봉사기들은 인증된 사용자들에게만 접근을 국한시키며 봉사요청을 인증할수 있어야 한다. 이 환경에서 워크스테이션은 망봉사들에 대한 자기의 사용자들을 정확히 확인한다고 신용할수 없다. 특히 다음의 3가지 위협들이 존재한다.

- 사용자는 특정의 워크스테이션에 접근하여 그 워크스테이션에서 조작되는 다른 사용자로 가장할수 있다.
- 사용자는 워크스테이션의 망주소를 변경하여 다른 워크스테이션에서 보낸 요청이 원래의 워크스테이션에서 온것처럼 속일수 있다.
- 사용자는 교환정보를 도청하고 반복공격을 리용하여 어떤 봉사에 대한 입구를 얻거나 조작을 혼란시킬수 있다.

이런 경우에 권한이 없는 사용자는 접근허가를 받지 않고도 봉사나 자료를 얻을수 있다. Kerberos는 매개 봉사기들에 인증규약들을 장비하기보다는 봉사기들에 사용자들을 인증시키고 사용자들에게는 봉사기들을 인증시키는 중앙인증봉사기를 제공한다. 이 책에서 서술된 다른 대부분의 인증방식들과는 달리 Kerberos는 공개열쇠암호를 리용하지 않고 전통암호만을 리용한다.

현재 두개 판본의 Kerberos들이 일반적으로 쓰이고 있다. 그러나 아직은 판본 4[MILL88, STEI88]도 널리 쓰이고 있다. 판본 4의 보안결함의 일부를 바로 잡아 창조한 판본 5[KOHL94]가 인터네트규격(RFC1510)으로서 발표되였다.

이 절에 대한 소개를 Kerberos취급법에 대한 간단한 론의로부터 시작하자. Kerberos가 복잡하므로 먼저 판본 4에서 쓰이는 인증규약부터 소개한다.이것은 포착하기 힘든 보안위협들을 조종하는데 요구되는 일부 세부들을 고찰함이 없이도 Kerberos전

략의 본질을 파악할수 있게 한다. 마지막으로 판본 5를 평가한다.

계기

만일 사용자들에게 망접속이 없는 개인용콤퓨터들이 제공된다면 사용자의 자원과 파일들은 매 개인용콤퓨터들을 물리적으로 안전하게 함으로써 보호할수 있다. 그대신 사용자들이 중심적인 시분할체계를 봉사 받을 때 그 체계는 보안을 담보해야 한다. 조작체계는 사용자신원에 기초한 접근조종방법들을 실시할수 있으며 사용자들을 확인하는데 등록가입수속(logon procedure)을 리용할수 있다.

아직 이 씨나리오들중의 어느것도 전형적인것으로 되지 못하고 있다. 보다 일반적인 것은 사용자워크스테이션(의뢰기)들과 분배 또는 중심화된 봉사기들로 구성되는 분배방식이다.

이 환경에서 보안에 대한 세가지 취급법들을 볼수 있다.

1. 매개 개별적의뢰기에 의존하여 그 사용자들의 신원을 확인하며 매개 봉사기에 의존하여 사용자신원(ID)에 기초한 보안방략을 강화한다.
2. 의뢰기체계가 자기자신을 확인할것을 봉사기들에 요구하는데 이로써 그 사용자의 신원과 관련한 의뢰체계는 신용된다.
3. 사용자에게는 매개 봉사들에 대하여 신원을 증명할것이 요구된다. 또한 봉사기들도 자기들의 신원을 증명할것을 의뢰기들에 요구한다.

모든 체계들이 어떤 한개 기관의 소유로 되여 조작되는 작은 닫긴환경에서는 첫번째 또는 두번째 방략이면 충분할것이다. 그러나 다른 콤퓨터들에 대한 망련결들이 지원되는 열린환경에서의 세번째 취급방법에서는 봉사기에 보관된 사용자정보나 자원들을 보호할 필요가 있다. 이 세번째 방법은 Kerberos에 의해 지원된다. Kerberos는 분배된 의뢰기/봉사기방식을 가정하고 하나 또는 그이상의 Kerberos봉사기들을 리용하여 인증봉사를 제공한다.

Keveros에 대하여 처음으로 공개된 보고에서는 다음의 요구조건들이 지적되였다[STEI88].

- **안전성:** 망도청자는 사용자로 가장하여도 필요한 정보를 얻을수 없다. 더 일반적으로 Kerberos는 적이 약한 고리를 찾지 못할만큼 충분히 강해야 한다.
- **신뢰성:**접근조종에서 Kerberos에 의존하는 모든 봉사들에 대한 Kerberos봉사의 유효성의 부족은 곧 지원되는 봉사들의 유효성부족을 의미한다. 따라서 Kerberos는 고도로 믿음직해야 하며 한 체계가 다른 체계를 지원할수 있는 분포된 봉사기방식을 리용하여야 한다.
- **투명성:** 리상적으로 볼 때 사용자는 통과암호를 입력하지 않는 한 인증이 진행되고 있다고 생각하지 말아야 한다.
- **증가성:** 체계는 많은 의뢰기들과 봉사기들을 지원할수 있어야 한다. 이것은 모듈분산방식을 제기한다.

이 요구조건들을 제공하는데서 Kerberos의 총적인 방식은 5장에서 론의되는 니드함(Needham)과 스레더(Schroeder)에 의한 방식[NEED78]에 기초한 규약을 리용하는 제3자에 의한 인증봉사인것이다. 이것은 의뢰기들과 봉사기들이 Kerberos를 믿고 그것들의 호상인증을 Kerberos가 조정한다는 의미에서 신용된다. 리상적으로 인증봉사는

Kerberos봉사기 그자체가 안전하면 안전하다고 말할수 있다.

Kerberos판본 4

Kerberos판본 4는 규약에서 인증봉사를 제공하는데 DES를 리용한다. 총체적인 규약을 보면 거기에 포함된 많은 요소들의 필요성을 밝히는것이 어렵다는것을 알수 있다. 따라서 Project Athena의 빌 브리안트(Bill Bryant)가 사용한 방략[BRVA88]을 써서 몇개의 가설적대화들에 먼저 류의하여 충분한 규약을 만든다. 매번 진행된 론의들에서는 앞의 론의들에서 로출된 보안약점들을 막기 위하여 추가적인 조작들을 부가한다.

그 규약을 설명한 다음 판본 4의 일부 다른 측면들을 본다.

간단한 인증대화

보호대책이 없는 망환경에서 의뢰기가 봉사기에 봉사를 신청하는 경우를 보자. 명백한 보안위협은 가장하는것이다. 공격자는 다른 의뢰기로 가장하여 봉사기들로부터 권한을 부여 받지 못한 특권들을 얻을수 있다. 이 위협을 막기 위하여 봉사기들은 봉사를 요구하는 의뢰기들의 신원을 확인할수 있어야 한다. 매 봉사기들이 의뢰기/봉사기대화의 과제를 떠맡을것이 요구되는데 열린환경에서 이것은 매개 봉사기들에 상당한 부담을 준다.

다른 방법은 모든 사용자들의 통과암호를 알고 이것들을 중심자료기지에 보관하는 인증봉사기를 리용하는것이다. 이밖에도 인증봉사기(AS)는 매개 봉사기와 유일한 비밀열쇠를 공유한다. 이 열쇠들은 물리적 또는 다른 안전한 방법으로 배포된다. 다음의 가정적인 대화를 고찰하자.

(1) C→AS: $ID_C \| P_C \| ID_V$
(2) AS→C: Ticket
(3) C→V: $ID_C \| Ticket$
$Ticket = E_{K_V}[ID_C \| AD_C | ID_V]$

여기서

C : 의뢰기
AS : 인증봉사기
V : 봉사기
ID_C : C에서 사용자의 식별자
ID_V : V의 식별자
P_C : C에서 사용자의 통과암호
AD_C : C의 망주소
K_V : AS와 V가 공유한 비밀암호열쇠
$\|$: 련결

이 씨나리오에서 사용자는 워크스테이션에 등록가입하고 봉사기 V에 접근을 요청한다. 사용자의 워크스테이션에서 의뢰모듈 C는 사용자의 통과암호를 요구하고 다음 AS에 사용자의 ID, 봉사기의 ID 및 사용자의 통과암호를 포함하는 통보문을 보낸다. AS는 자료기지를 검사하고 사용자가 자기의 ID에 대한 정확한 통과암호를 제공하였는가와 이 사용자가 봉사기 V에 대한 접근을 허락 받았는가를 알아 본다. 만일 두가지 검사가 다

통과되면 AS는 사용자가 확인된것으로 수락하고 봉사기에 이 사용자가 확인되였다는것을 알려야 한다. 이를 위해 AS는 사용자의 ID, 망주소 및 봉사기의 ID를 포함하는 증명서(ticket)를 작성한다. 이 증명서는 AS와 이 봉사기가 공유한 비밀열쇠로 암호화된다. 다음 이 증명서를 C에 돌려 보낸다. 증명서가 암호화되였으므로 C나 적에 의해 변경되지 않는다.

이 증명서로 C는 V에 봉사를 신청할수 있다. C는 자기의 ID와 증명서를 포함하는 통보문을 V에 보낸다. V는 그 증명서를 복호하여 그 사용자 ID가 통보문의 암호화되지 않은 사용자 ID와 같다는것을 검증한다. 만일 이 두개가 일치하면 봉사기는 사용자가 확인된것으로 간주하고 요청한 봉사를 허락한다.

통보문 (3)의 매 요소들이 중요하다. 증명서는 암호화되므로 변경이나 위조를 막는다. 봉사기의 ID(ID_V)는 증명서에 포함되며 따라서 봉사기는 증명서를 정확히 복호했다는것을 검증할수 있다. 이 증명서가 C의 이름으로 발행된것이라는것을 지적하기 위하여 해당 ID를 증명서에 포함시킨다. 마지막으로 AD_C는 다음의 위협을 막는다. 적은 통보문 (2)에서 전송되는 증명서를 도청한 다음 이름 ID_C를 리용하여 형식 (3)의 통보문을 다른 워크스테이션으로부터 전송할수 있다. 봉사기는 사용자 ID와 일치하는 정당한 증명서를 받고 다른 워크스테이션의 그 사용자에 대한 접근을 허락할수 있다. 이 공격을 막기 위하여 AS는 증명서에 신청한 의뢰기의 망주소를 포함시킨다. 이때 그 증명서는 처음에 그것을 신청한 워크스테이션으로부터 전송될 때에만 정당하다.

더 안전한 인증대화

앞에서 본 씨나리오가 열린망환경에서 인증에 대한 일부 문제들은 해결하지만 문제거리는 아직 남아 있다. 특히 두가지가 중요하다. 첫째로, 사용자가 통과암호를 넣기하는 회수를 최소화하는것이다. 매개 증명서가 한번만 리용될수 있다고 가정한다. 만일 사용자 C가 아침에 워크스테이션에 등록가입하고 우편봉사기에서 자기의 우편을 검사하려고 한다면 C는 우편봉사기에 증명서를 주기 위해 통과암호를 제공하여야 한다. 만일 C가 하루에 우편을 여러번 검사하려고 한다면 매번 통과암호를 재넣기할것이 요구된다. 증명서들이 재리용가능하다고 하면 문제가 달라 진다. 단일등록가입방식에서 워크스테이션은 우편봉사기증명서를 받은 다음 그것을 보관하고 사용자가 우편봉사기에 여러번 접근할 때 그것을 사용한다.

그러나 이 방식에서 사용자가 매번 다른 봉사를 받기 위하여 새로운 증명서를 요구해야 하는 경우가 있을수 있다. 만일 사용자가 인쇄봉사기, 우편봉사기, 파일봉사기 등에 접근하려고 한다면 매 접근의 첫 경우에 새로운 증명서가 요구되며 따라서 사용자가 통과암호를 넣을것을 요구한다.

두번째 문제는 앞의 씨나리오에 통과암호(통보문 1)의 평문전송이 포함되였다는것이다. 도청자는 통과암호를 가로 채여 피해자에게 접근가능한 임의의 봉사를 리용할수 있다.

이 추가적인 문제들을 해결하기 위하여 증명서-허가봉사기(TGS)라고 하는 새로운 봉사기와 평문통과암호를 피하는 방식을 도입한다. 새롭지만 아직 가설적인 씨나리오는 다음과 같다.

사용자등록가입당 한번:

(1) C→AS: $ID_C \| ID_{tgs}$

(2) AS→C: $E_{K_c}[Tickt_{tgs}]$

봉사의 형태당 한번:

(3) C→TGS: $ID_C \| ID_V \| Ticket_{tgs}$

(4) TGS→C: $Ticket_V$

봉사대화당 한번:

(5) C→V: $ID_C \| Ticket_V$

$Ticket_{tgs} = E_{K_{tgs}} [ID_C \| AD_C | ID_{tgs} \| TS_1 \| Lifefime_1]$

$Ticket_V = E_{K_V} [ID_C \| AD_C | ID_V \| TS_2 \| Lifefime_2]$

새로운 봉사를 위해 TGS는 AS에 확인된 사용자들에게 증명서들을 발행한다. 따라서 사용자는 먼저 AS로부터 증명서-허가증명서($Ticket_{tgs}$)를 요청한다. 이 증명서는 의뢰모듈에 의하여 사용자의 워크스테이션에 보관된다. 사용자가 매번 새로운 봉사에 대한 접근을 요구할 때마다 의뢰기는 자기자신을 확인하는 증명서를 써서 TGS에 문의한다. 그러면 TGS는 개개의 봉사에 대하여 어떤 증명서를 허락한다. 의뢰기는 매개 봉사-허가증명서들을 보관하며 그것으로 개개의 봉사들이 요청될 때마다 봉사기의 사용자를 확인하는데 리용한다. 이 방식을 구체적으로 고찰하자.

1. 의뢰기는 AS에 자기의 사용자 ID를 TGS봉사를 리용하기 위한 요구를 지적하는 TGS ID와 함께 보내여 사용자의 이름으로 된 증명서-허가증명서를 요청한다.
2. AS는 사용자의 통과암호로부터 유도되는 열쇠를 암호화된 증명서로 응답한다. 이 응답이 의뢰기에 전달되면 의뢰기는 사용자에게 그의 통과암호를 지적하여 열쇠를 생성하고 들어 온 통보문의 복호를 시도한다. 만일 정확한 통과암호가 제공되면 그 증명서는 성공적으로 복호된다.

정당한 사용자만이 통과암호를 알고 있으므로 그 증명서를 복호할수 있다. 즉 평문에 통과암호를 전송하지 않고 통과암호를 리용하여 Kerberos로부터의 신임장을 얻는다. 그 증명서자체는 ID, 사용자의 망주소 및 TGS의 ID로 구성된다. 이것은 첫번째 씨나리오에 해당된다. 기본원리는 의뢰기가 여러가지 봉사-허가증명서들을 요구하는데 이 증명서를 리용할수 있다는것이다. 그러므로 증명서-허가증명서는 재리용될수 있다. 그러나 적이 증명서를 도청하여 그것을 리용하는 일이 있어서는 안된다. 다음과 같은 씨나리오를 고찰해 보자. 즉 적은 증명서를 도청하고 사용자가 자기의 워크스테이션을 등록탈퇴할 때까지 기다릴수 있다. 적은 그 워크스테이션에 대한 접근을 얻거나 피해자와 같은 망주소를 가지도록 가장한다. 적은 그 증명서를 리용하여 TGS를 완전히 속일수 있다. 이것을 막기 위하여 증명서는 그것이 발행된 날자와 시간을 가리키는 시간도장(시간표시)과 그 증명서의 유효시간의 길이를 가리키는 생명주기를 포함한다. 따라서 의뢰기는 재리용가능한 증명서를 가지며 매번 새로운 봉사요구때마다 통과암호를 주지 않아도 된다. 마지막으로 증명서-허가증명서는 AS와 TGS만이 아는 비밀열쇠로 암호화된다는데 방점이 있다. 이것은 증명서의 변경을 막는다. 그 증명서는 사용자의 통과암호에 기초한 열쇠로 재암호화된다. 이것은 증명서가 신원이 확인되는 정확한 사용자에 의해서만 복호될

수 있다는것을 담보한다.

의뢰기가 증명서-허가증명서를 가지므로 임의의 봉사기에 대한 접근을 단계 3과 4에서 얻을수 있다. 즉

3. 의뢰기는 사용자의 이름으로 봉사-허가증명서를 요청한다. 이를 위하여 의뢰기는 사용자의 ID, 요구하는 봉사의 ID 및 증명서-허가증명서를 포함하는 통보문을 TGS에 전송한다.
4. TGS는 들어 온 증명서를 복호하고 자기의 ID에 의해 복호결과를 검증한다. TGS는 생명주기가 끝나지 않았다는것을 확인한다. 다음 사용자 ID와 망주소를 들어 온 정보와 비교하여 사용자를 확인한다. 만일 사용자가 V에 대한 접근을 허락 받으면 TGS는 요청 받은 봉사에 대한 접근을 허락하는 증명서를 발행한다.

봉사-허가증명서는 증명서-허가증명서와 같은 구조를 가진다. 사실 TGS도 봉사기이므로 TGS에 의뢰기를 확인시키고 응용봉사기에 의뢰기를 확인시키는데서 같은 요소들이 리용된다는것을 알수 있다. 또한 증명서는 증명서시간도장과 생명주기를 포함한다. 만일 사용자가 후에 같은 봉사에 접근하려고 하면 의뢰기는 간단히 이전에 얻은 봉사-허가증명서를 리용할수 있으므로 통과암호재넣기를 하지 않아도 된다. 증명서는 변경을 막기 위해 TGS와 봉사기들만이 아는 비밀열쇠(K_V)로 암호화된다.

마지막으로 의뢰기는 개개의 봉사-허가증명서로 단계 5에 해당한 봉사에 대한 접근을 얻을수 있다.

5. 의뢰기는 사용자의 이름으로 봉사에 대한 접근을 요구한다. 이를 위하여 의뢰기는 봉사기에 사용자의 ID와 봉사-허가증명서를 포함하는 통보문을 전송한다. 봉사기는 증명서의 내용들을 리용하여 확인한다.

이 새로운 씨나리오는 사용자대화당 한개의 통과암호질문과 사용자통과암호보호의 두 요구를 만족한다.

판본 4의 인증대화

앞의 씨나리오에 의해 첫 시도에 비하여 보안은 높아 졌지만 아직 두가지 문제가 남아 있다. 첫 문제의 중심은 증명서-허가증명서와 관련한 생명주기이다. 만일 그 생명주기가 매우 짧으면(즉 분단위) 사용자는 자주 통과암호를 응답해 주어야 한다. 만일 생명주기가 길면(즉 시간단위) 적에게 그 증명서를 리용당할수 있는 기회를 줄수 있다. 적은 망에서 도청하여 증명서-허가증명서를 복사한 다음 정당한 사용자가 등록탈퇴하기를 기다린다. 다음 적은 정당한 사용자의 망주소를 기만하고 단계 3의 통보문을 TGS에 보낸다. 이로써 적은 정당한 사용자만이 접근할수 있었던 자원과 파일들에 대한 접근을 할수 있게 된다.

마찬가지로 만일 적이 봉사-허가증명서를 획득하여 그것의 생명주기가 끝나기전에 쓰면 해당 봉사에 대한 접근을 할수 있다.

따라서 만족시켜야 할 다음의 추가적인 조건들을 부가할수 있다. 망봉사(TGS나 응용봉사)는 증명서를 사용하는 사람이 바로 그 증명서를 받은 사람과 같다는것을 증명할수 있어야 한다.

두번째 문제는 봉사기들이 자기를 사용자들에게 확인시킬 필요가 있을수 있다는것이다. 이러한 인증이 없으면 적은 봉사기에로의 통보문이 다른 곳으로 가게 하도록 구성을

위조할수 있다. 그러면 가짜봉사기가 실지 봉사기로서 동작하는 위치에 있게 되고 사용자로부터의 모든 정보를 획득하며 사용자에 대한 진짜봉사를 부정할수 있다.

이 문제들을 차례로 조사하고 표 11-1을 참조한다. 이 표는 실지의 Kerberos규약을 보여 준다.

먼저 증명서-허가증명서를 획득하는 문제와 증명서표식자가 그 증명서를 받은 의뢰기와 갈다는것을 결정할 필요를 설명하자. 위협은 적이 증명서를 훔치여 그것의 생명주기가 끝나기전에 리용하는것이다. 이 문제를 해결하기 위하여 AS가 의뢰기와 TGS에 안전한 방법으로 비밀정보를 제공한다고 하자. 그러면 의뢰기는 다시 안전한 방법으로 비밀정보를 보여 줌으로써 TGS에게 자기의 신원을 증명할수 있다. 이것을 실현하는 효과적인 방법은 안전한 정보로서 암호열쇠를 리용하는것이다. 즉 이것을 Kerberos에서는 대화열쇠라고 부른다.

표 11-1의 ㄱ는 대화열쇠로 알려 진 열쇠를 배포하는 기술을 보여 준다. 앞에서와 마찬가지로 의뢰기는 AS에 TGS에 대한 접근을 요구하는 통보문을 보낸다. AS는 증명서를 포함하는 사용자의 통과암호(Kc)로부터 도출되는 열쇠로 암호화된 통보문으로 응답한다.

표 11-1. Kerberos판본 4의 통보문교환의 개략

ㄱ) 인증봉사교환: 증명서-허가증명서를 얻는다.
(1) $C \rightarrow AS$: $ID_C \| ID_{tgs} \| TS_1$ (2) $AS \rightarrow C$: $E_{Kc}[K_{C,tgs} \| ID_{tgs} \| TS_2 \| Lifefime_2 \| Ticket_{tgs}]$ $Ticket_{tgs} = E_{K_{tgs}}[K_{C,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifefime_2]$
ㄴ) 증명서-허가봉사교환: 봉사-허가증명서를 얻는다.
(3) $C \rightarrow TGS$: $ID_V \| Ticket_{tgs} \| Authenticator_C$ (4) $TGS \rightarrow C$: $E_{Kc,tgs}[K_{C,V} \| ID_V \| TS_4 \| Ticket_V]$ $Ticket_{tgs} = E_{K_{tgs}}[K_{C,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifefime_2]$ $Ticket_V = E_{K_V}[K_{C,V} \| ID_C \| AD_C \| ID_V \| TS_4 \| Lifefime_4]$ $Authenticator_C = E_{Kc,tgs}[K_{C,V} \| ID_C \| AD_C \| TS_3]$
ㄷ) 의뢰기 /봉사기인증교환: 봉사를 얻는다.
(5) $C \rightarrow K$: $Ticket_V \| Authenticator_C$ (6) $K \rightarrow C$: $E_{K_{C,V}}[TS_5+1]$ (상호인증을 위하여) $Ticket_V = E_{K_V}[K_{C,V} \| ID_C \| AD_C \| ID_V \| TS_4 \| Lifefime_4]$ $Authenticator_C = E_{K_{C,V}}[ID_C \| AD_C \| TS_5]$

암호화된 통보문도 역시 대화열쇠인 $K_{C,tgs}$의 복사를 포함한다. 여기서 첨자들은 이

것이 C와 TGS에 대한 대화열쇠라는것을 의미한다. 이 대화열쇠가 K_C로 암호화된 통보문의 내부에 있으므로 그 사용자의 의뢰기만이 그것을 읽을수 있다. 같은 대화열쇠가 증명서에 포함되여 있는데 그것은 TGS만이 읽을수 있다. 이리하여 대화열쇠는 C와 TGS에게 안전하게 전달된다.

먼저 몇가지 정보들이 대화의 첫 단계에 부가된다는것을 강조한다. 통보문 (1)은 그 통보문이 시기적절하다는것을 AS가 알도록 시간도장을 포함한다. 통보문 (2)는 C에 접근가능한 형식의 증명서의 몇개 요소들을 포함한다. 이것은 C가 이 증명서는 TGS에 대한것이라는것을 확인하게 하며 그것의 소멸시간을 알게 한다.

증명서와 대화열쇠로 《무장》되면 C는 TGS에 접근할 준비가 된다. 전과 같이 C는증명서에 요청된 봉사의 ID(표 11-1의 ㄴ)와 통보문 (3)을 함께 포함하는 통보문을 TGS에게 보낸다. 더우기 C는 인증자를 전송하는데 그것은 C의 사용자의 ID와 주소 및 시간도장을 포함한다. 재리용가능한 증명서와는 달리 인증자는 오직 한번만 리용되게 되며 매우 짧은 생명주기를 가진다. TGS는 AS와 공유한 열쇠로 그 증명서를 복호할수 있다. 이 증명서는 사용자 C에게 대화열쇠 $K_{C,tgs}$가 제공된다는것을 알려 준다. 사실상 그 증명서는 《C만이 $K_{C,tgs}$를 사용할수 있다》라는것을 의미한다. TGS는 대화열쇠를 사용하여 인증자를 복호한다. 다음 TGS는 증명서의 이름과 들어 온 통보문의 망주소로 인증자의 이름과 주소를 검사할수 있다. 모두 일치하면 TGS는 그 증명서의 송신자가 그것의 진짜 주인이라는것을 확신한다.

인증자는 《시각 TS_3에서 나는 이 결과에 의해 $K_{C,tgs}$를 사용한다》는것을 의미한다. 그 증명서는 누구의 신원도 증명하지 못하며 다만 열쇠들을 안전하게 배포하기 위한 한가지 방법이라는것을 주의해 둔다. 의뢰기의 신원을 증명하는것이 인증자이다. 인증자는 한번만 쓰이며 짧은 생명주기를 가지므로 적이 후날 재연하기 위해 증명서와 인증자를 둘 다 훔칠 위협은 없어 지게 된다.

TGS로부터의 대답은 통보문 (4)에서 통보문 (2)의 형식을 따른다. 그 통보문은 TGS와 C가 공유한 대화열쇠로 암호화되며 C와 봉사기 V사이에 공유된 대화열쇠, V의 ID 및 그 증명서의 시간도장을 포함한다. 증명서자체는 같은 대화열쇠를 포함한다.

이때 C는 V에 재리용가능한 봉사-허가증명서를 가진다. 통보문 (5)에서와 같이 C는 이 증명서를 제출할 때 인증자도 보낸다. 봉사기는 그 증명서를 복호하여 대화열쇠를 재현할수 있으며 인증자를 복호한다.

만일 호상인증이 요구되면 봉사기는 표 11-1의 통보문 (6)에서처럼 응답할수 있다. 봉사기는 인증자로부터 1만큼 증가되고 그 대화열쇠로 암호화된 시간도장의 값을 돌려준다. C는 이 통보문을 복호하여 증가된 시간도장을 재현할수 있다. 그 통보문이 대화열쇠로 암호화되였으므로 C는 그것이 V에 의하여 만들어 진것이라는것을 확신한다. 그 통보문의 내용은 C에게 이것이 낡은 응답의 재현이 아니라는것을 확인시킨다.

마지막으로 이 과정을 끝내면서 의뢰기와 봉사기는 비밀열쇠를 공유한다. 이 열쇠는 후에 둘사이에서 통보문들을 암호화하는데 쓰이거나 그러한 목적에서 새로운 우연대화열쇠를 교환하는데 쓸수 있다.

표 11-2는 Kerberos규약의 매 요소들에 대한 정당성을 개괄하고 그림 11-1은 그 동작을 간단히 보여 준다.

Kerberos범위와 다중커버리(Kerberi)

Kerberos봉사기, 여러대의 의뢰기들과 응용봉사기들로 구성되는 옹근-봉사 Kerberos환경은 다음의것을 요구한다.

표 11-2. Kerberos 판본 4규약의 요소들에 대한 리론적기초

ㄱ) 인증봉사교환	
통보문 (1):	의뢰기는 증명서-허가증명서를 요구한다.
ID_C:	이 의뢰기사용자의 신원을 AS에 알린다.
ID_{tgs}:	AS에 사용자가 TGS에 대한 접근을 요구한다는것을 알린다.
TS_1:	AS가 의뢰기의 박자와 자기의 박자가 동기화된다는것을 검증할수 있게 한다.
통보문 (2):	AS는 증명서-허가증명서를 돌려 준다.
E_{K_C} :	암호화는 사용자의 통과암호에 기초하며 AS와 의뢰기가 통과암호를 검증하게 하며 통보문 (2)의 내용을 보호한다.
$K_{C,\ tgs}$:	의뢰기에 접근가능한 대화열쇠의 복사: 항구적인 열쇠공유를 하지 않고 의뢰기와 TGS사이에 안전한 교환을 허락하기 위하여 AS가 창조한다.
ID_{tgs} :	이 증명서가 TGS를 위한것이라는것을 확인한다.
TS_2 :	의뢰기에 이 증명서가 발행된 시간을 알린다.
$Lifetime_2$:	의뢰기에 이 증명서의 생명주기를 알린다.
$Ticket_{tgs}$:	TGS에 접근하기 위하여 의뢰기가 사용하는 증명서
ㄴ) 증명서-허가봉사교환	
통보문 (3):	의뢰기는 봉사-허가증명서를 요청한다.
ID_V :	TGS에 사용자가 봉사기 V에 대한 접근을 요구한다는것을 알린다.
$Ticket_{tgs}$:	이 사용자가 AS에 의해 인증되였다는것을 TGS에 확인시킨다.
$Authenticator_C$:	증명서를 확인하기 위하여 의뢰기에 의해 생성
통보문 (4):	TGS는 봉사-허가증명서를 돌려 준다.
$E_{K_{C,\ tgs}}$:	C와 TGS만이 공유한 열쇠: 통보문 (4)의 내용을 보호한다.
$K_{C,\ tgs}$:	의뢰기에 접근가능한 대화열쇠의 복사: 의뢰기와 봉사기가 항구적인 열쇠를 공유하지 않아도 그것들사이의 안전한 교환을 허용할수 있도록 TGS에 의해 창조된다.
ID_V :	이 증명서가 봉사기 V를 위한것이라는것을 확인한다.
TS_4 :	의뢰기에 이 증명서가 발행된 시간을 알린다.
$Ticket_V$:	봉사기 V에 접근하기 위하여 의뢰기가 사용하는 증명서
$Ticket_{tgs}$:	사용자가 통과암호를 다시 넣지 않아도 재리용가능하다.
$E_{K_{tgs}}$:	증명서는 가로채기를 막기 위하여 AS와 TGS만이 아는 열쇠로 암호화된다.
$K_{C,\ tgs}$:	TGS에 접근가능한 대화열쇠의 복사: 인증자를 복호하는데 쓰이므로 인증증명서로 된다.
ID_C :	이 증명서의 합법적인 소유자를 가리킨다.
AD_C :	그 증명서를 처음에 요구하지 않은 다른 워크스테이션에서의 증명서사용을 막는다.

표계속

ID_{tgs}:	봉사기가 증명서를 정확히 복호했다는것을 보증한다.
TS_2:	TGS에게 이 증명서가 발행된 시간을 알려 준다.
$Lifetime_2$:	소멸된 증명서의 재리용을 막는다.
$Authenticator_C$:	TGS에게 그 증명서제출자가 그것을 발행한 의뢰기와 같다는것을 확인한다. 재연을 막기 위하여 매우 짧은 생명주기를 가진다.
$E_{Kc,tgs}$:	인증자는 가로채기를 막기 위하여 의뢰기와 TGS만이 아는 열쇠로 암호화된다.
ID_C:	증명서를 인증하기 위하여 증명서의 ID에 일치하여야 한다.
AD_C:	증명서를 인증하기 위하여 증명서의 주소와 일치하여야 한다.
TS_2:	TGS에 이 인증자가 생성된 시간을 알려 준다.
ㄷ) 의뢰기/봉사기인증교환	
통보문 (5)	의뢰기는 봉사를 요구한다.
$Ticket_V$:	봉사기에 이 사용자가 AS에 의해 인증된다는것을 보증한다.
$Authenticator_C$:	증명서의 정당성을 검증하기 위하여 의뢰기에 의해 생성
통보문 (6)	의뢰기에 대한 봉사기의 임의의 인증
$E_{K_{C,V}}$:	C에게 이 통보문이 V로부터 온것이라는것을 보증한다.
TS_5+1:	C에게 그것이 이전 응답의 재연이 아니라는것을 보증한다.
$Ticket_V$:	재리용가능하므로 의뢰기가 같은 봉사기에 매번 접근할 때마다 TGS로부터 새 증명서를 요구할 필요가 없다.
E_{K_V}:	증명서는 가로채기를 막기 위하여 TGS와 봉사기만이 아는 열쇠로 암호화된다.
$K_{C,V}$:	의뢰기에로의 접근가능한 대화열쇠의 복사; 인증자를 복호하는데 쓰이며 따라서 증명서를 인증한다.
ID_C:	이 증명서의 정당한 소유자라는것을 가리킨다.
AD_C:	처음에 증명서를 요구했던 워크스테이션이 아니면 그 증명서를 리용하지 못하게 한다.
ID_V:	증명서를 정확히 복호했다는것을 봉사기에 보증한다.
TS_4:	봉사기에 이 증명서가 발행된 시간을 알려 준다.
생명주기:	증명서가 소멸된후 재연을 막는다.
$Authenticator_C$:	증명서제출자가 그 증명서를 발행한 의뢰기와 같다는것을 봉사기에 보증한다. 재연을 막기 위해 매우 짧은 생명주기를 가진다.
$E_{K_{C,V}}$:	인증자는 가로채기를 막기 위해 의뢰기와 봉사기만이 아는 열쇠로 암호화된다.
ID_C:	증명서를 보증하려면 그의 ID와 일치하여야 한다.
AD_C:	증명서를 보증하려면 그의 주소와 일치하여야 한다.
TS_5:	이 인증자가 발생된 시간을 봉사기에 알려 준다.

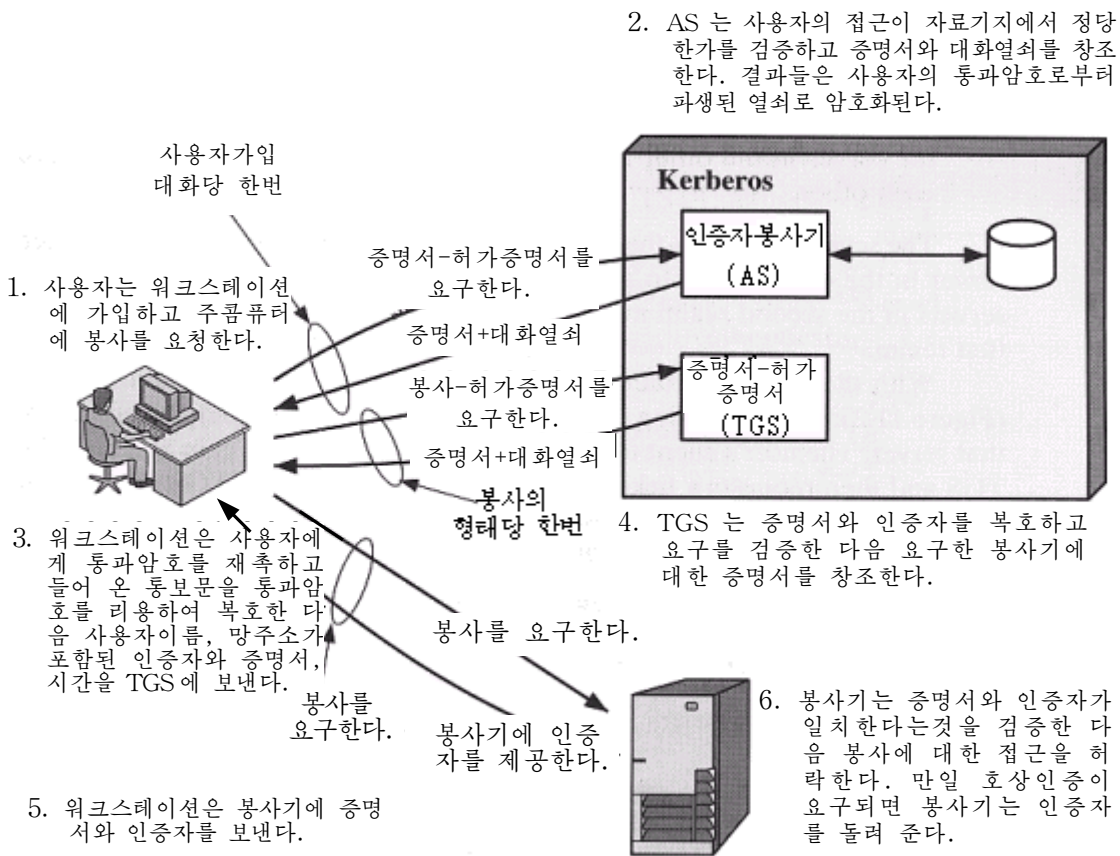



그림 11-1. Kerberos의 개괄

1. Kerberos봉사기는 자기의 자료기지에 사용자ID(UID)와 모든 관계자들의 하쉬 통과암호들을 가지고 있어야 한다. 모든 사용자들은 Kerberos봉사기에 등록되여 있다.
2. Kerberos봉사기는 매개 봉사기와 비밀열쇠를 공유하여야 한다. 모든 봉사기들은 Kerberos봉사기에 등록되여 있다.

이러한 하나의 환경을 범위(realm)라고 한다. 서로 다른 관리기관들 내부의 의뢰기들과 봉사기들로 이루어 지는 망들은 일반적으로 서로 다른 범위들을 구성한다. 즉 하나의 관리령역의 사용자들과 봉사기들이 다른 곳의 Kerberos봉사기에 등록되는것은 현실적이 못되거나 관리에 적합하지 않다. 그러나 한 범위에서 사용자들이 다른 범위에 있는 봉사기들에 대한 접근을 요구할수 있으며 일부 봉사기들은 다른 범위의 인증된 사용자들에게 봉사를 제공할수 있다.

Kerberos는 이러한 범위간인증을 지원하는 꾸밈새를 제공한다. 두개의 범위에서 범위간(inter-realm)인증을 지원하기 위하여 다음의 세번째 조건이 부가된다.

3. 매개의 호상작용하는 범위에서 Kerberos봉사기는 다른 범위의 봉사기와 비밀열쇠를 공유한다. 이 두개의 Kerberos봉사기들은 서로 대방에게 등록된다.

이 방식은 어떤 범위에 있는 Kerberos봉사기가 다른 범위에 있는 Kerberos봉사기를 믿고 그것의 사용자들을 인증할것을 요구한다. 더우기 두번째 범위에 관계하는 봉사기들은 첫 범위에 있는 Kerberos봉사기를 믿으려 해야 한다.

이런 기본규칙들을 리용하여 꾸밈새를 다음과 같이 서술할수 있다(그림 11-2). 다른 범위안의 봉사기에 대한 봉사를 원하는 사용자는 그 봉사기에 대한 증명서를 요구한다. 그 사용자의 의뢰기는 보통 절차에 따라 국부TGS에 대한 접근을 얻기 위해 일반수속을 준수하며 다음 어떤 원격의 TGS(다른 범위의 TGS)를 위한 증명서-허가증명서를 요구한다. 다음 의뢰기는 원격의 TGS의 범위에 있는 요구하는 봉사기를 위한 봉사-허가증명서를 얻기 위해 원격의 TGS에 신청할수 있다.

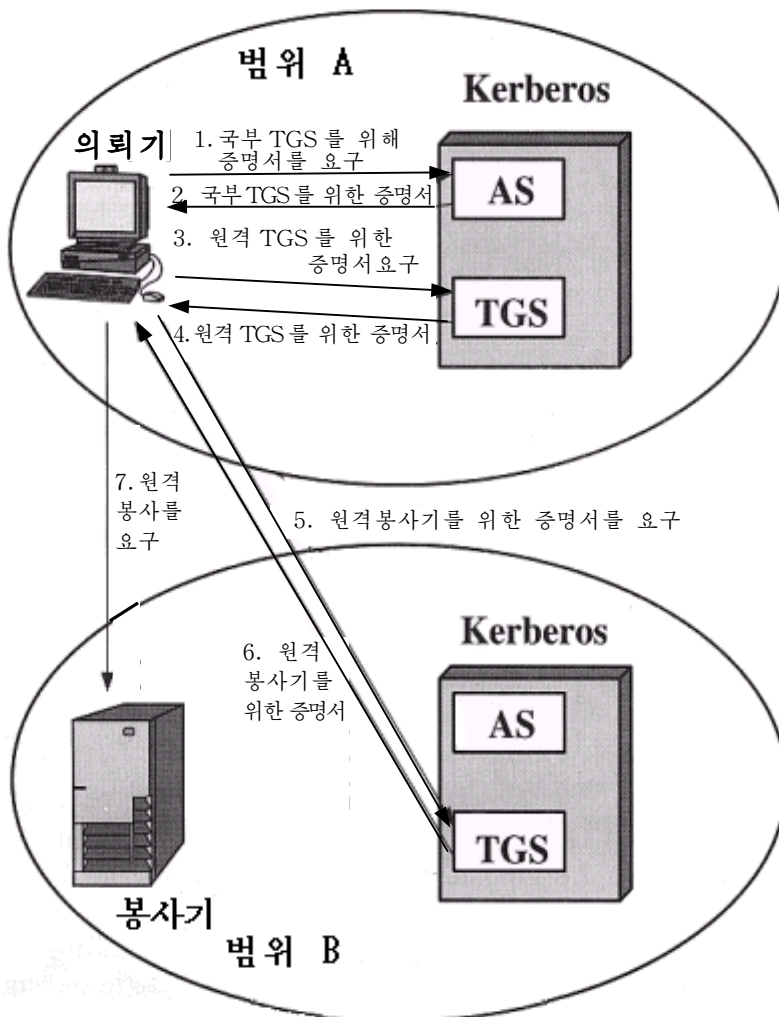



그림 11-2. 다른 범위의 봉사를 요구한다

그림 11-2에서 보여 준 교환들의 상세한 내용은 다음과 같다(표 11-1을 비교하라).:

(1) C→AS: $ID_c \| ID_{tgs} \| TS_1$
(2) AS →C: E_{K_c} [$K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}$]
(3) C→TGS: $ID_{tgsrem} \| Ticket_{tgs} \| Authenticator_c$
(4) TGS →C: $E_{K_{c,tgs}}$ [$K_{c,tgsrem} \| ID_{tgsrem} \| TS_4 \| Ticket_{tgsrem}$]
(5) C→TGS_{rem}: $ID_{vrem} \| Ticket_{tgsrem} \| Authenticator_c$
(6) TGS→ C: $E_{K_{c,tgsrem}}$ [$K_{cvrem} \| ID_{vrem} \| TS_b \| Ticket_{vrem}$]
(7) C→V_{rem}: $Ticket_{vrem} \| Authenticator_c$

원격봉사기(V_{rem})에 제출된 증명서에 그 사용자가 본래 인증되였던 범위가 지적되였다. 봉사기는 원격의 요구를 들어 주겠는가 안주겠는가를 선택한다.

앞의 수법에서 제기되는 하나의 문제는 그것이 많은 범위들에 대하여 그리 효과적이지 못하다는것이다. 만일 N개의 범위들이 있으면 매개 Kerberos범위들이 다른 모든 범위들과 호상작용할수 있도록 N(N-1)/2번의 안전한 열쇠교환이 있어야 한다.

Kerberos판본5

Kerberos의 판본 5는 RFC1510에 서술되였는데 판본4[KOHL94]에 비해 많이 개선되였다. 론의에 앞서 판본 4와 판본 5의 차이들을 제시한 다음 판본5의 규약을 보자.

판본 4와 판본 5의 차이

판본 5는 두개의 령역 즉 환경적인 결함들과 기술부족측면에서 판본 4의 제한성들을 극복한것이다. 매개 령역에서 개선된것들만 간단히 개괄하자.

판본 4의 Kerberos들은 Project Athena환경에서 사용하기 위해 개발되였으며 따라서 일반목적의 요구는 충분히 만족시키지 못한다. 이것은 다음 환경적결함들을 발로시킨다.

1. **암호체계의존성:** 판본 4는 DES를 리용하도록 되여 있다. DES에 대한 반출제한이나 DES의 강도에 대한 의심들이 그러한 문제이다. 판본 5에서는 임의의 암호기술을 리용할수 있도록 암호문에 암호형식별자가 붙어 다닌다. 암호화열쇠들에는 일정한 형식과 길이의 꼬리표를 달아 주어 서로 다른 알고리듬들에 같은 열쇠를 쓸수 있게 하고 또 주어 진 알고리듬의 여러가지 변종들의 명세들을 허락한다.
2. **인터네트규약의존성:** 판본 4는 IP주소를 사용하도록 되여 있다. ISO망주소와 같은 다른 주소형식들과는 융통성이 없다. 판본 5의 망주소들에 형식과 길이가 붙으므로 임의의 망주소형을 쓸수 있게 하였다.
3. **통보문의 바이트정돈:** 판본 4에서는 어떤 통보문의 송신자가 그자신이 선정한 바이트(a byte ordering)를 써서 제일 아래 주소에서 최소 유효자료비트나 최대유효자료비트를 나타내도록 통보문에 꼬리표들을 달아 준다. 이 기술은 제정된 협약에 따르지 않고 쓰인다. 판본 5에서는 모든 통보문구조들을 ASN.1 (Abstract Syntax Notation One)과 BER(Basic Encoding Rules)를 리용하여 정의하는데 이것들은 명백한 바이트배렬을 제공한다.
4. **증명서생명주기:** 판본 4에서 생명주기값은 5분을 단위로 8비트량으로 코드화되였다. 따라서 표시할수 있는 최대생명주기는 $2^8 \times 5=1280$분 즉 21시간을 조금 넘는

다. 이것은 일부 응용(실례로 실행 전기간 유효한 Kerberos신임장을 요구하는 긴시간-실행모의)들에서 불충분하다. 판본 5에서는 증명서들에 명백한 시작시간과 끝시간을 지적함으로써 증명서들에 생명주기를 임의로 할수 있게 한다.

5. **인증발송:** 판본 4는 하나의 의뢰기에 대해 발행된 신임장이 다른 가입자에게 발송되거나 다른 의뢰기에 의해 리용되지 못하게 한다. 이 기능은 의뢰기가 어떤 봉사기에 접근하고 그 봉사기가 그 의뢰기의 이름으로 다른 봉사기에 접근할수 있게 한다. 실례로 의뢰기가 인쇄봉사기에 요구를 제출하면 그 인쇄봉사기는 의뢰기의 접근신임장을 리용하여 단일봉사기에서 의뢰기파일에 접근한다. 판본 5는 이 기능을 제공한다.
6. **내부범위인증:** 판본 4에서 N개 범위들의 호상작용성은 앞에서 서술한것처럼 약 N^2개의 Kerberos-대-Kerberos관계들을 요구한다. 판본 5는 보다 적은 수의 관계들을 요구하는 방법을 지원한다.

이 환경적제한들외에도 판본 4의 규약자체에는 기술적으로 부족한 점들이 있다. 그 대부분이 문헌[BELL90]에 지적되여 있는데 판본 5에서는 이것들에 주의가 돌려 지고 있다. 그 부족점들은 다음과 같다.

1. **2중암호:** 표 11-1(통보문 2와 4)에서 의뢰기들에 제공된 증명서들은 두번은 목적봉사기의 비밀열쇠로, 다음은 다시 의뢰기가 아는 비밀열쇠로 암호화된다. 두번째 암호화는 필요 없으며 계산량적으로 랑비이다.
2. **PCBC암호:** 판본 4의 암호화에서는 증식암호블로크련쇄(PCBC-propagating cipher block chaining)로 알려 진 DES의 비표준방식이 리용된다. 이 방식은 암호문블로크들의 호상교체를 포함하는 공격에 약하다는것이 알려 져 있다[KOHL89]. PCBC는 암호조작의 부분으로서 완정성검사를 제공한다. 판본 5는 표준CBC방식을 암호화에 쓸수 있게 하는 명시적완정성꾸밈새를 제공한다.
3. **대화열쇠(Session keys):** 매개 증명서들은 대화열쇠를 포함하는데 그것은 의뢰기가 그 증명서와 관련한 봉사에 보내 지는 인증자를 암호화하는데 쓰인다. 또한 대화열쇠는 의뢰기와 봉사기가 그 대화시간동안 통과하는 통보문을 보호하기 위하여 리용한다. 그러나 같은 증명서가 어떤 특정의 봉사기로부터 봉사를 얻기 위해 반복하여 쓰일수 있으므로 적이 낡은(이전의)대화의 증명서로 의뢰기나 봉사기에 통보문들을 재연할 위험이 있다. 판본 5에서는 의뢰기와 봉사기가 부분대화열쇠를 약속할수 있는데 그것은 오직 그 한번의 접속을 위해서만 쓰는것이다. 의뢰기에 의한 새로운 접근은 새로운 부분대화열쇠의 리용에 기인한다.
4. **통과암호공격:** 두개의 판본이 다 통과암호공격에는 약하다. AS로부터 의뢰기에로의 통보문에는 의뢰기의 통과암호에 기초한 열쇠로 암호화된 자료가 있다. 적은 이 통보문을 획득하여 각이한 통과암호들을 써가면서 그것을 복호하려고 한다. 시험복호결과가 의미 있는 형식이면 적은 의뢰기의 통과암호를 발견하고 이어 그것을 리용하여 Kerberos로부터의 인증신임장을 얻을수 있다. 이것이 15장에서 서술한것과 같은 형태의 통과암호공격인데 같은 종류의 대책을 취할수 있다. 판본 5는 사전인증(preauthenticaton)으로 알려 진 꾸밈새를 제공하는데 그것도 통과암호공격을 더 어렵게는 하지만 그것들을 완전히 막지는 못한다.

판본 5인증대화

표 11-3에 기본판본 5의 대화를 개괄하였다. 이것은 판본 4(표 11-1)와 비교하여 잘 설명된다.

먼저 인증봉사교환을 고찰한다. 통보문 (1)은 증명서-허가증명서에 대한 의뢰기요구이다. 이전처럼 거기에는 사용자와 TGS의 ID가 포함된다. 다음의 새로운 요소들이 부가된다.

- Realm: 사용자의 범위를 가리킨다.
- Options : 돌아 오는 증명서에 일정한 표식들이 설정될것을 요구하는데 쓰인다.
- Times : 의뢰기가 증명서에서 다음의 시간설정을 요구하는데 리용된다.
  - ¤ From : 요구 받은 증명서의 필요한 시작시간
  - ¤ till: 요구 받은 증명서의 요구되는 소멸시간
  - ¤ rtime: 요구 받은 교체소멸시간
- Nonce: 통보문 (2)에서 응답이 신선하고 적에 의하여 재연되지 않았다는것을 확인하기 위해 반복되는 우연값

표 11-3. Kerberos판본 5통보문교환의 개괄

ㄱ) 인증봉사교환: 증명서-허가증명서를 얻는다.
(1) $C \rightarrow AS$: $Options \| ID_C \| Realm_C \| ID_{tgs} \| Times \| Nonce_1$ (2) $S \rightarrow C$: $Realm_C \| ID_C \| Ticket_{tgs} \| E_{Kc}[K_{C,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}]$ $Ticket_{tgs} = E_{K_{tgs}}[Flags \| K_{C,tgs} \| Realm_C \| ID_C \| AD_C \| Times]$
ㄴ) 증명서-허가봉사교환: 봉사-허가증명서를 얻는다.
(3) $C \rightarrow TGS$: $Options \| ID_V \| Times \| Nonce_2 \| Ticket_{tgs} \| Athenticator_C$ (4) $TGS \rightarrow C$: $Realm_C \| ID_C \| Ticket_V \| E_{C,tgs}[K_{C,V} \| Times \| Nonce_2 \| Realm_V \| ID_V]$ $Ticket_{tgs} = E_{K_{tgs}}[Flags \| K_{C,tgs} \| Realm_C \| ID_C \| AD_C \| Times]$ $Ticket_V = E_{K_v}[Flags \| K_{C,V} \| Realm_C \| ID_C \| AD_C \| Times]$ $Authenticator_C = E_{Kc,tgs}[ID_C \| Realm_C \| TS_1]$
ㄷ) 의뢰기/봉사기인증교환: 봉사를 얻는다.
(5) $C \rightarrow TGS$: $Options \| Ticket_V \| Authenticator_C$ (6) $TGS \rightarrow C$: $E_{K_{C,V}}[TS_2 \| Subkey \| Seq\#]$ $Ticket_V = E_{K_V}[Flags \| K_{c,v} \| Realm_C \| ID_C \| AD_C \| Times]$ $Authnticator_C = E_{K_{C,V}}[ID_C \| Realm_C \| TS_2 \| Subkey \| Seq\#]$

통보문 (2)는 의뢰기에 대한 정보를 확인하는 증명서-허가증명서와 사용자의 통과암호에 기초한 암호화열쇠에 의하여 암호화된 블로크를 돌려 준다. 이 블로크는 의뢰기와 TGS사이에 쓰이는 대화열쇠, 통보문 (1)에 렬거된 시간들, 통보문 (1)로부터의 한번쓰기정보 및 TGS확인정보를 담는다. 증명서 그자체는 대화열쇠, 의뢰기를 확인하는 정보, 요구 받은 시간정보들과 이 증명서의 상태 및 요구 받은 선택항목들을 반영하는 표식들을 포함한다. 이 기발들은 판본5에 대한 새로운 중요한 기능을 시사한다. 먼저 판본5규약의 전반적구조를 보면 다음과 같다.

우선 **증명서-허가봉사교환**을 판본 4와 판본 5를 비교하여 고찰하자. 두 판본들에서 통보문 (3)에는 인증자, 증명서 및 요구하는 봉사의 이름을 포함한다. 그의 판본 5는 증명서가 요구시간들과 그 증명서에 대한 선택항목들, 한번쓰기정보 그리고 통보문(1)의것들과 류사한 기능들을 가진것들을 모두 포함한다. 인증자 그 자체는 판본4에서 쓰던것과 본질적으로 같다.

통보문(4)는 통보문(2)와 같은 구조를 가지면서 의뢰기가 요구하는 정보와 증명서를 함께 돌려 주는데 통보문은 의뢰기와 TGS가 공유한 대화열쇠로 암호화된다.

결국 판본 5에는 의뢰기/봉사기인증교환(client/server authentication exchange)에 대한 몇가지 새로운 특성들이 있다. 통보문(5)에서 의뢰기는 호상인증을 요구하는것을 선택항목으로 요구할수 있다. 인증자는 다음과 같은 몇개의 새로운 마당들을 포함한다.

- **부분열쇠:** 이 특정의 응용대화를 보호하기 위하여 리용되는 암호열쇠에 대한 의뢰기의 선택. 이 마당이 빠지면 증명서($K_{c,v}$)로부터의 대화열쇠가 리용된다.
- **렬번호:** 이 대화기간에 의뢰기에 보내지는 통보문들을 위하여 봉사기가 리용하는 시작렬번호를 서술하는 선택마당. 통보문들은 재연들을 탐지하기 위하여 렬번호화될수 있다.

만일 호상인증이 요구되면 봉사기는 통보문 (5)로써 응답한다. 이 통보문은 인증자로부터의 시간도장을 포함한다. 판본4에서 시간도장은 하나씩 증가하였다. 판본 5에서는 통보문들의 형식화특성이 적이 정확한 암호열쇠를 알지 못하고서는 통보문 (6)을 창조할수 없는것이므로 이것이 필요 없다. 부분열쇠마당이 존재하면 통보문(5)의 부분열쇠마당을 증가한다. 선택렬번호마당은 의뢰기가 리용하는 시작렬번호를 명기한다.

증명서기발

판본 5에서 증명서들에 포함된 기발마당은 판본 4에서의 경우에 비해 확장된 기능을 지원한다. 표 11-4는 증명서에 도입되는 기발들을 개괄한다.

INITIAL기발은 이 증명서를 TGS가 아니라 AS가 발행했다는것을 가리킨다. 의뢰기가 TGS로부터 봉사-허가증명서를 요구하면 TGS는 AS로부터 얻은 증명서-허가증명서를 제공한다. 판본 4에서는 이것이 봉사-허가증명서를 얻는 방법이였다. 판본5는 의뢰기가 AS로부터 봉사-허가증명서를 직접 얻을수 있는 부가적능력을 제공한다. 그 리용은 다음과 같다. 통과암호-변경봉사기와 같은 봉사기는 그 의뢰기의 통과암호가 최근에 검사되였다는것을 알려고 할수 있다.

PRE-AUTHENT기발들이 설정되면 이것은 AS가 첫 요구(통보문1)를 받았을 때 증명서를 발행하기전에 그 의뢰기를 인증했다는것을 가리킨다. 이 사전인증의 정확한 형식은 아직 정의되지 않았다. 실례로서 판본 5의 MIT실장은 시간도장사전인증을 암호화하였다. 사용자가 증명서를 얻으려 하면 의뢰기의 통과암호기초의 열쇠로 암호화된 우연섞임물, 판본번호 및 사건표시를 포함하는 사전인증블로크를 AS에 보내야 한다. AS는

표 11-4. Kerberos판본 5의 기발들

INITIAL	이 증명서는 AS규약을 리용하여 발행되였으나 증명서-허가증명서에 기초하여 발행되지는 않았다.
PRE-AUTHENT	초기인증동안 증명서가 발행되기전에 KDC가 의뢰기를 확인하였다.
HW-AUTHENT	초기인증에 리용되는 규약은 지정된 의뢰기만이 소유하게 될 하드웨어의 리용을 요구하였다.
RENEWABLE	이 증명서가 후에 소멸될 교체증명서를 얻는데 리용할수 있다는것을 TGS에 알려 준다.
MAY-POSTDATE	사후년기의 증명서가 이 증명서-허가증명서에 기초하여 발행되였다는것을 TGS에 알려 준다.
POSTDATED	이 증명서가 사후년기되였다는것을 가리킨다. 즉 말단봉사기는 인증시간마당을 검사하여 첫 인증이 진행된 시간을 알수 있다.
INVALID	이 증명서는 쓸수 없으며 리용전에 KDC에 의하여 유효성이 검증되여야 한다.
PROXIABLE	TGS에 다른 망주소를 가진 새로운 봉사-허가증명서가 제출된 증명서에 기초하여 발행될수 있다는것을 알려 준다.
PROXY	이 증명서가 대리인이라는것을 가리킨다.
FORWARDABLE	다른 망주소를 가진 새로운 증명서-허가증명서가 이 증명서-허가증명서에 기초하여 발행될수 있다는것을 TGS에 알려 준다.
FORWARDED	이 증명서가 전송되였거나 전송된 증명서-허가증명서를 포함하는 인증에 기초하여 발급되였다는것을 지정한다.

그 블로크를 복호하고 그 사전인증블로크의 시간도장이 정당한 시간내에 들어 가지 않으면 증명서-허가증명서를 돌려 보내지 않는다. 다른 가능성은 사전인증되는 통보문들에 포함되는 계속적으로 변하는 통과암호를 생성하는 간단한 카드의 리용이다. 그 카드가 생성하는 통과암호들은 사용자의 통과암호에 기초하지만 요컨대 임의의 통과암호들이 리용되도록 카드에 의해 변환될수 있다. 이것은 쉽게 추측되는 통과암호들에 기초한 공격을 막는다. 만일 스마트카드나 그와 류사한 장치가 리용되면 이것은 HW-AUTHENT기발로 지적된다.

증명서가 긴 생명주기를 가지면 그것을 도적 맞히고 적이 리용할수 있는 가능성이 있다. 반대로 짧은 생명주기가 위협을 줄이는데 리용된다면 새로운 증명서들을 획득하는데 간접비용이 든다. 증명서-허가증명서의 경우에 의뢰기는 사용자의 비밀열쇠를 보관해야 하거나(이것은 틀림없이 모험이다.) 사용자에게 반복적으로 통과암호를 요구하여야 한다.

타협방식은 재생증명서들의 리용이다. RENEWABLE기발설정을 가지는 증명서는 두개의 소멸시간을 포함하는데 하나는 이 증명서에 고유한것이고 다른 하나는 소멸시간의 제일 마지막 허용값이다.

의뢰기는 새로운 소멸시간을 가진 TGS로 그것을 제출하여 RENEW된 증명서를 얻을수 있다. 만일 새로운 시간이 제일 마지막 허용값의 한계내에 들어 가면 TGS는 새로운 대화시간과 어떤 이후의 특정한 소멸시간을 가지는 새로운 증명서를 발행한다.

이 기구의 우점은 TGS가 도적 맞혔다고 신고된 증명서를 재생하는것을 거절할수

있는것이다.

의뢰기는 AS가 MAY-POSTDATE기발설정을 가지는 증명서-허가증명서를 제공할것을 요구할수 있다. 그리고 이 증명서를 리용하여 TGS로부터 POSTDATE와 INVALID로 기발설정된 증명서를 요구할수 있다. 이 방식은 증명서를 주기적으로 요구하는 봉사기에서 긴 묶음일감을 실행하는데 쓸모 있다. 의뢰기는 이 대화기간에 확장시간값들을 가지는 많은 증명서들을 한번에 얻을수 있다. 첫 증명서를 내놓고는 모두 처음에는 무효이다. 실행이 새로운 증명서가 요구되는 시점에 도달할 때 의뢰기는 적당한 증명서를 유효화할수 있다. 이 수법에서 의뢰기는 봉사허가증명서를 얻는데 자기의 증명서-허가증명서를 반복하여 리용하지 말아야 한다.

판본 5에서 봉사기는 다른 봉사기로부터의 봉사를 위해 의뢰기의 신임장과 특권을 효과적으로 채택함으로써 그 의뢰기의 대리인처럼 활동할수 있다. 만일 의뢰기가 이 꾸밈새를 리용하려고 한다면 PROXIABLE기발설정을 가지는 증명서-허가증명서를 요구한다. 이 증명서가 TGS에 제출되면 TGS는 서로 다른 망주소들을 가지는 봉사-허가증명서를 발행할데 대하여 허락 받는다. 즉 이후의 증명서는 그것의 PROXY기발설정을 가진다. 이러한 증명서를 받는 응용은 그것을 접수하거나 또는 검열추적을 제공하기 위해 부가적인 인증을 요구할수 있다.

대리인개념은 더 힘 있는 전송절차의 일부 경우이다. 만일 증명서가 FORWARDABLE기발로써 설정되면 TGS는 서로 다른 망주소와 FORWARDED기발설정을 가지는 증명서-허가증명서를 요구자에게 발행할수 있다. 다음 이 증명서는 원격TGS에 제출된다. 이 능력은 의뢰기가 다른 범위에 있는 봉사기에 대한 접근을 매개 Kerberos가 다른 모든 범위의 Kerberos봉사기들과 비밀열쇠를 유지하지 않고서도 얻을수 있게 한다. 실례로 범위들은 계층적으로 구성될수 있다. 그러면 의뢰기는 나무를 따라 공통마디로 갔다가 목표범위로 들어 갈수 있다. 방문의 매 단계는 경로상에서 증명서-허가증명서를 다음의 TGS에로의 전송을 포함한다.

11.2 X.509등록부인증봉사

ITU-T권고의 X.509는 등록부봉사를 정의하는 권고들의 X.500계렬의 부분이다. 등록부는 사실상 사용자들에 대한 정보자료기지를 보관하는 봉사기 또는 봉사기들의 분포된 모임이다. 정보에는 망주소로 사영한 사용자의 이름이나 기타 사용자와 관련한 속성이나 자료가 들어 있을수 있다.

X.509는 사용자들에 대하여 X.500등록부에 의한 인증봉사준비를 위한 구성을 정의한다. 등록부는 6장에서 서술된 형태의 공개열쇠증명서들의 저장소로서 봉사할수 있다. 매개 증명서들은 사용자의 공개열쇠들을 포함하며 신용된 증명국의 비밀열쇠로 서명된다. 또한 X.509는 공개열쇠증명서의 리용에 기초한 교차인증규약을 정의한다.

X.509에서 정의된 증명서구조와 인증규약들이 여러 응용들에서 쓰이므로 X.509는 중요한 표준이다. 실례로 X.509의 증명서형식은 S/MIME(12장), IP Security(13장), SSL/TLS 및 SET(14장)에서 리용된다.

X.509는 1988년에 처음으로 공개되였다. 그 표준은 후에 [IANS90]과 [MTC90]에서 서술된 일부 보안문제와 관련하여 계속 개정되였으며 그 개정판이 1993년에 공개되였다. 세번째 판본은 1995년에 초안이 작성되였다.

X.509는 공개열쇠암호와 수자서명에 기초하고 있다. 표준은 특정한 알고리듬을 사

용하지 않지만 RSA에 의거한다. 수자서명방식은 하쉬함수를 리용하는것으로 가정하였다. 게다가 표준은 특정한 하쉬알고리듬의 사용을 지정하지 않는다. 1988년판에서는 선택된 하쉬알고리듬의 해설을 주었다. 후에도 이 알고리듬은 안전하지 못하다는것이 밝혀지고 있으며 1993년의 개정판에서 삭제되였다.

증명서

X.509방식의 기본 핵은 매 사용자들에 대한 공개열쇠증명서이다. 이 사용자증명서들은 신용되는 증명국(CA: Certification Authority)이 발급하며 CA나 사용자에 의해 등록부에 배치된다. 등록부봉사기자체는 공개열쇠의 발급이나 증명기능의 창조를 감당하지 않고 다만 사용자가 증명서를 얻기 위하여 쉽게 접근할수 있는 장소를 제공한다.

그림 11-3의 ㄱ에 증명서의 일반형식을 보여 준다. 그것은 다음의 요소들을 포함한다.

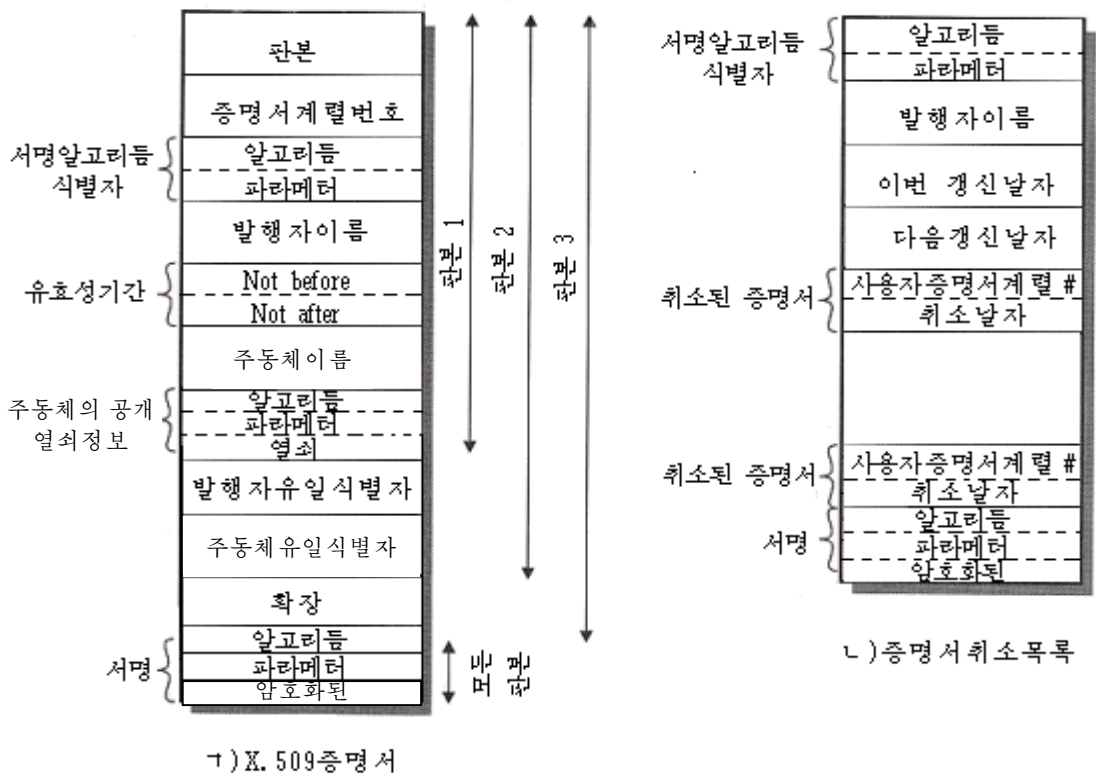



그림 11-3. X.509형식

- **판본:** 증명서형식의 련속적인 판본들을 구별한다. 즉 기정판은 판본 1이다. 만일 Initiator Unique Identifier나 Subject Unique Identifier가 있으면 그 값은 판본 2이다. 만일 하나 또는 그 이상의 확장들이 있으면 그 값은 판본 3이다.
- **계렬번호:** 발행되는 CA내에서 유일한 옹근수로서 증명서와의 관련이 모호한것

- **서명알고리듬식별자:** 모든 관련되는 파라메터들과 함께 증명서를 서명하는데 리용되는 알고리듬. 이 정보는 증명서의 마지막 서명마당에서 반복되므로 이 마당은 임의의 리용이 있다고 해도 적다.
- **발행자이름:** 이 증명서를 발급하고 서명한 CA의 X.500이름
- **유효성기간:** 두개의 날자로 이루어 진다. 그 증명서가 유효한 처음과 마지막시간으로 구성된다.
- **대상이름:** 이 증명서가 위임되는 사용자의 이름. 즉 이 증명서는 대응하는 비밀열쇠를 가지는 대상의 공개열쇠를 확인한다.
- **대상의 공개열쇠정보:** 다른 모든 관련파라메터들과 함께 이 열쇠가 리용되는 알고리듬의 식별자와 대상의 공개열쇠
- **발행자유일식별자:** 다른 실체들에서 X.509이름이 재리용되는 경우에 발행CA를 유일하게 식별하기 위하여 리용하는 선택적비트렬마당
- **대상유일식별자:** 다른 실체들에서 X.509이름이 재리용되는 경우에 그 대상을 유일하게 식별하기 위하여 리용하는 선택적비트렬마당
- **확장:** 하나 또는 그 이상의 확장마당들의 모임. 확장은 판본 3에 부가되는데 이 절의 뒤에서 론의한다.
- **서명:** 증명서의 다른 모든 마당들을 포함한다. 즉 이것은 CA의 비밀열쇠로 암호화된 다른 부분들의 하쉬코드를 포함한다. 이 부분은 서명알고리듬식별자를 포함한다.

유일식별자마당들은 판본 2에 부가되여 시간외 대상 및/또는 발행자이름들의 가능한 재리용을 조정한다. 이 마당들은 그리 리용되지 않는다.

표준은 다음의 표시를 리용하여 증명서를 정의한다.

$$CA<<A>>=CA\{U, SN, AI, CA, T_A, A, A_P\}$$

여기서

Y<<X>> = 증명국 Y가 발행한 사용자 X의 증명서

Y{I}= Y에 의한 I의 서명. 이것은 암호화된 하쉬코드가 덧달린 I로 구성한다.

CA는 증명서에 자기의 비밀열쇠로 서명한다. 만일 대응한 공개열쇠가 이 사용자에게 알려 지면 사용자는 CA가 서명한 증명서가 정당한가를 검증할수 있다. 이것은 그림 8-5의 ㄷ에서 설명된 전형적인 수자서명이다.

사용자의 증명서얻기

CA에 의해 생성된 사용자증명서는 다음의 특성들을 가진다.

- CA의 공개열쇠에 접근할수 있는 임의의 사용자는 보증된 사용자공개열쇠를 회복할수 있다.
- 증명국이 아닌 다른것들은 그 증명서를 변경시킬수 없다.

증명서들이 위조불가능하므로 등록부가 그것들을 특별히 보호할 필요없이 등록부에

배치할수 있다.

만일 모든 사용자들이 같은 CA에 예약하면 그 CA에 대한 일반적인 신임이 있는것이다. 모든 사용자증명서들은 접근을 위해 등록부에 배치할수 있다. 또한 사용자는 자기의 증명서를 다른 사용자들에게 직접 전송할수 있다. 어느 경우든 B가 A의 증명서를 소유하기만 하면 B는 A의 공개열쇠로 암호화한 통보문이 도청으로부터 안전하며 A의 비밀열쇠로 서명된 통보문도 위조불가능하다는 확신을 가질수 있다.

만일 큰 사용자공동체가 있으면 모든 사용자들이 같은 CA에 예약한다고는 할수 없다. 증명서들에 서명하는것도 CA이므로 매 참가자(사용자)들은 CA의 공개열쇠에 의한 복사를 가지고 서명들을 검증하여야 한다. 이 공개열쇠는 사용자들이 증명서에 대하여 믿음을 가지도록 절대적으로 안전한(완정성과 인증성에 대하여) 방법으로 매개 사용자들에게 제공되여야 한다. 따라서 사용자들이 많으면 여러개의 CA들이 있는것이 더욱 현실적일것이다.

이제 A가 증명국 X_1 로부터 증명서를 받고 B는 증명국 X_2로부터 증명서를 받았다고 가정하자. 만일 A가 X_2의 공개열쇠를 확고히 알수 없으면 A는 B의 증명서를 쓸수 없다. A는 B의 증명서를 읽을수 있지만 그 서명을 확인할수 없다. 그러나 만일 두개의 CA들이 자기들의 공개열쇠를 안전하게 교환하면 다음의 절차에 의해 A는 B의 공개열쇠를 얻을수 있다.

1. A는 등록부로부터 X_1이 서명한 X_2의 증명서를 얻는다. A는 X_1의 공개열쇠를 확고히 알고 있으므로 자기의 증명서로부터 X_2의 공개열쇠를 얻을수 있으며 그것을 증명서의 X_1의 서명을 리용하여 검증할수 있다.
2. 다음에 A는 등록부로 돌아 가서 X_2이 서명한 B의 증명서를 얻는다. 이때 A는 X_2의 공개열쇠에 대한 신용되는 복사를 가지므로 서명을 검증하고 B의 공개열쇠를 안전하게 얻을수 있다.

A는 증명서련쇄를 리용하여 B의 공개열쇠를 얻는다. X.509의 표기법에서 이 련쇄는 다음과 같이 표시할수 있다.

$$X_1 \ll X_2 \gg X_2 \ll B \gg$$

같은 방식으로 B는 A의 공개열쇠를 거꿀련쇄로 얻을수 있다.

$$X_2 \ll X_1 \gg X_1 \ll A \gg$$

이 방식은 두개의 증명서의 련쇄에만 극한되지 않는다. 임의로 긴 CA들의 로정을 련쇄을 따라 창조할수 있다. N개의 요소들을 가지는 련쇄는

$$X_1 \ll X_2 \gg X_2 \ll X_3 \gg \cdots X_N \ll B \gg$$

로 표현된다. 이 경우에 련쇄(X_i, X_{i+1})에서 매 CA들의 쌍은 서로에 대하여 증명서들을 만들어야 한다.

CA들사이의 이 모든 증명서들은 등록부에 나타나며 사용자는 다른 사용자의 공개열쇠증명서에로의 경로에 그것들이 어떻게 련결되여 가는가를 알아야 한다. X.509는 탐색이 간단하도록 CA들이 계층적으로 배렬된것을 제안한다.

그림 11-4에 이러한 계층의 실례를 보여 주었다. 련결된 원은 CA들속에서 계층적 관계를 의미하며 또 그와 련결된 칸들은 매개 CA가입을 위하여 등록부에 유지되는 증명서들을 가리킨다. 매개 CA에 대한 등록부가입은 두가지 형태의 증명서들을 포함한다.

- **앞방향증명서:** 다른 CA들에 의해 생성된 X의 증명서들
- **반대방향증명서:** 다른 CA들의 증명서들인 X에 의해 생성된 증명서들

이 실례에서 사용자 A는 등록부로부터 다음의 증명서들을 알고 B에로의 증명로정을 설정할수 있다.

X＜＜W＞＞W＜＜V＞＞V＜＜Y＞＞Y＜＜Z＞＞Z＜＜B＞＞

A가 이 증명서들을 얻으면 그는 증명로정들을 차례로 헤쳐 보면서 B의 공개열쇠의 확실한 복사를 발견할수 있다. 이 공개열쇠를 써서 A는 B에게 암호화된 통보문을 보낸다.

만일 A가 B로부터 거꾸로 암호화된 통보문을 되받으려고 한다거나 B에게 보낸 통보문에 서명하려고 하면 B는 A의 공개열쇠를 요구한다. 다음의 증명과정으로부터 이것을 얻을수 있다.

Z＜＜Y＞＞Y＜＜V＞＞V＜＜W＞＞W＜＜X＞＞X＜＜A＞＞

B는 등록부로부터 이 증명서들의 모임을 얻을수 있다. 또한 A는 그것들을 B에게 초기통보문의 부분으로서 제공할수 있다.

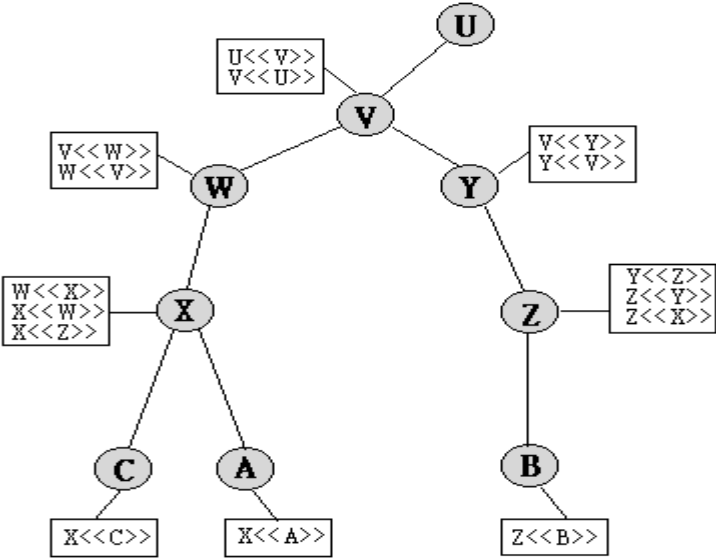



그림 11-4. X.509 계층(가설적인 실례)

증명서의 취소

그림 11-3으로부터 신용카드와 마찬가지로 매개 증명서는 유효기간을 포함한다는것을 상기하자. 일반적으로 새로운 증명서는 낡은것이 끝나기직전에 발행된다. 또한 다음의 리유들중의 하나로 하여 때때로 증명서의 유효기간이 끝나기전에 취소할것이 요구될수 있다.

1. 사용자의 비밀열쇠가 손상되였다고 인정된다.
2. 사용자는 이 CA에 의해 더 이상 보증되지 않는다.
3. CA의 증명서가 손상되였다고 가정된다.

매개 CA는 사용자들과 CA들에 발행된 증명서들을 포함하여 그 CA가 발행한 유효기간이 끝나지 않았지만 취소된 모든 증명서들로 이루어 지는 목록을 유지하여야 한다. 이 목록들은 등록부에 기입된다.

등록부에 게시된 매개 증명서취소목록(CRL-certificate revocation list)은 발행자에 의해 서명되며 발행자의 이름, 그 목록이 창조된 날자, CRL이 발행된 날자 및 매개 취소된 증명서의 기입을 포함한다. 매개 입구자료(entry)는 증명서의 계렬번호와 취소날자로 구성된다. 계렬번호들은 하나의 CA에서 유일하므로 그 증명서를 확인하는데 충분하다.

사용자가 통보문으로 증명서를 받으면 그 증명서가 취소되였는가 어떤가를 결정해야 한다. 사용자는 증명서를 받을 때마다 등록부를 검사한다. 등록부탐색과 관련한 지연(및 가능한 비용)을 피하기 위하여 사용자는 증명서, 목록 또는 취소된 증명서들의 국부캐시를 유지할수 있다.

인증절차

X.509는 여러가지 응용들을 교차적으로 리용하기 위하여 창조된 3가지 인증절차들을 포함한다. 이 모든 절차들은 공개열쇠서명을 리용한다. 두 대상이 서로 상대의 공개열쇠를 안다고 가정하자. 그림 11-5에 3가지 그 절차들을 보여 주었다.

한방향인증

한방향인증은 한 사용자(A)로부터 다른 사용자(B)에로의 정보의 단일전송을 포함한다. 다음과 같이 정의한다.

1. A의 확인과 통보문이 A에 의해 생성되였다는것
2. 그 통보문이 B에게 보내 진다는것
3. 그 통보문의 완정성과 초기성(여러번 보내 지지 않았다는것)

이 처리에서는 응답하는 실체의 정당성이 아니라 받아 들인 실체의 정당성만이 검증된다.

최소한 통보문은 시간도장 t_A, 한번쓰기정보 r_A 및 B의 식별자로 이루어 지며 A의 공개열쇠로 서명된다. 시간도장은 생성시간과 소멸시간으로 이루어 진다. 이것은 통보문의 지연된 배달을 막는다. 한번쓰기정보는 재연공격을 적발하는데 쓸수 있다. 그 값은 통보문의소멸시간내에서 유일해야 한다. 따라서 B는 한번쓰기정보가 소멸될 때까지 그것을 보관하고 같은 한번쓰기정보를 가진 새로운 통보문들을 거부할수 있다.

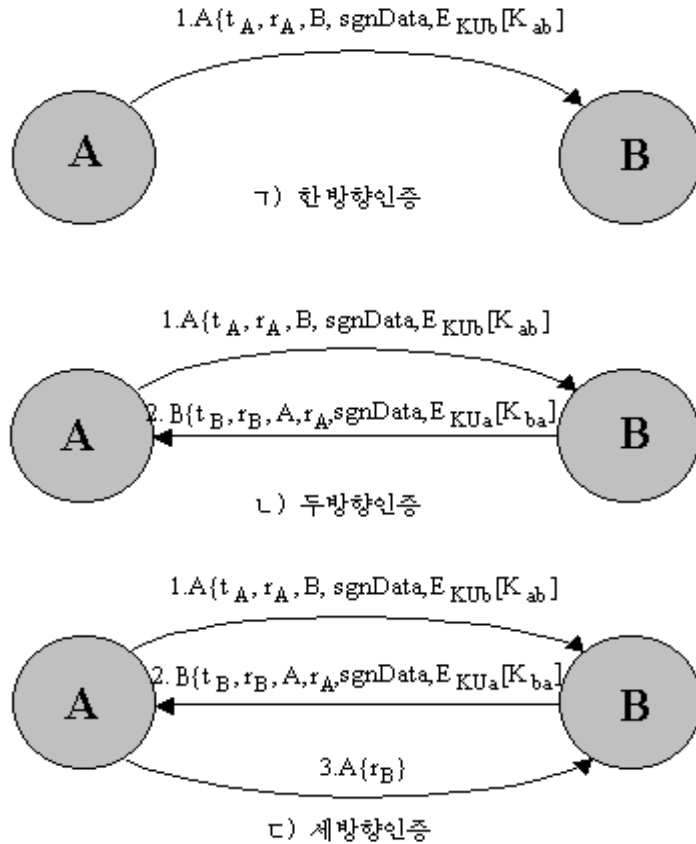



그림 11-5. X.509 강인증절차

순수한 인증에서 통보문은 단순히 B에 대한 신임장을 표현하는데 쓰인다. 통보문은 운반되는 정보를 포함할수도 있다. 이 정보 즉 서명자료는 서명의 범위에 포함되여 정당성과 완정성을 담보한다. 통보문은 또한 B의 공개열쇠로 암호화된 대화열쇠를 B에로 나르는데 쓸수 있다.

쌍방향인증

우에서 설명한 세 요소들외에 쌍방향인증은 다음의 요소들을 설정한다.

4. B의 신원과 응답통보문이 B에 의해 생성되였다는것
5. 그 통보문이 A에 가는것이라는것
6. 그 응답의 완정성과 초기성

이와 같이 쌍방향인증은 통신에서 다른 사람의 신원을 검증하는데 두개의 대상들을 다 허락한다.

응답통보문은 응답을 확인하기 위하여 A로부터의 한번쓰기정보를 포함한다. 그것은 또한 B가 생성한 시간도장과 한번쓰기정보도 포함한다. 앞에서와 마찬가지로 그 통보문

은 A의 공개열쇠로 암호화된 서명된 부가정보와 대화열쇠를 포함할수 있다.

세방향인증

세방향인증에는 A로부터 B에로의 마지막통보문이 포함되는데 거기에는 r_B의 서명된 복사가 들어 있다. 이 설계의 취지는 시간도장(timestamps)을 검사할 필요가 없다는것이다. 즉 두개의 한번쓰기정보들이 다 상대측에 의해 다시 되돌이되므로 매 측은 돌아온 한번쓰기정보을 검사하여 재연공격을 적발할수 있다. 이 방식은 동기화된 박자들을 쓸수 없을 때 필요하다.

X.509 판본 3

X.509 판본 2형식은 새로운 설계와 실장실현이 필요하게 된다고 본 정보들을 모두 나르지 못한다. 문헌[FORD95]에서는 판본 2에서 만족되지 않는 다음의 요구조건들을 소개한다.

1. 주동체마당은 공개열쇠사용자에게 열쇠소유자의 신원을 나르는데 적합치 않다. X.509이름들은 비교적 짧아서 사용자가 요구할수 있는 명백한 신원증명세부가 빠질수 있다.
2. 또한 주동체마당은 일반적으로 실체들을 인터네트전자우편주소, URL 또는 다른 일련의 인터네트관련의 확인으로 인식되는 많은 응용들에 대하여 불충분하다.
3. 보안방략정보를 지적할 필요가 있다. 이것은 보안응용이나 기능(IPSec와 같은)이 주어 진 방법에 대하여 X.509증명서를 관련시킬수 있다.
4. 특정의 증명서의 리용성에 제한들을 주어 결점이나 악의 있는 CA로부터 생길수 있는 피해를 제한할 필요가 있다.
5. 서로 다른 시간에 한 소유자가 리용할수 있는 서로 다른 열쇠를 확인하는것이 중요하다. 이 특성은 열쇠생명주기관리 특히 규칙적인 토대 또는 례외적인 정황밑에서 사용자나 CA들에 열쇠쌍을 갱신하는 능력을 지원한다.

규격의 개발자들은 고정된 형식에 부분들을 계속 부가하기보다 유연한 접근이 필요하다는것을 알았다. 따라서 판본 3은 판본 2형식에 첨부할수 있는 선택항목확장들을 포함한다. 매개 확장은 확장식별자, 위험성지시기 및 확장값으로 이루어 진다. 위험성지시기는 확장이 안전하게 무시될수 있는가를 가리킨다. 만일 지시기가 TRUE값을 가지고 실장이 확장을 인정하지 않으면 그것은 증명서를 정당하지 않은것으로 취급한다.

증명서확장들은 세개의 주요부류 즉 열쇠, 확인절차정보, 대상과 발행자특성 및 증명경로제한에 들어 간다.

열쇠와 방략정보

이 확장들은 대상과 발행자열쇠 및 증명서방략지적자들에 대한 추가적인 정보들을 담고 있다. 증명서방략은 공통된 보안요구를 가진 특정의 공동체와 부류 혹은 그 개개들에 증명서를 적용할수 있는가를 지적해 주는 규칙들의 일정한 모임이다. 실례로 방략은 주어 진 가격범위내에서 물건들을 매매하기 위한 전자자료교환업무의 인증에 적용할수 있다.

이 령역은 다음의것들을 포함한다.

- **증명국의 열쇠식별자:** 이 증명서나 CRL에 대한 서명을 검증하는데 리용되는 공개열쇠를 확인한다. 한 CA의 서로 다른 열쇠들을 구분할수 있게 한다. 이 마당의 한가지 리용은 CA열쇠쌍의 갱신을 조종하는것이다.
- **주동체의 열쇠식별자:** 검증되고 있는 공개열쇠를 확인한다. 주동체열쇠쌍을 갱신하는데 리용한다. 또한 대상은 여러개의 열쇠쌍들을 가지고 서로 다른 목적들에 서로 다른 증명서들을 리용할수 있다(즉 수자서명과 암호열쇠).
- **열쇠취급법:** 확인된 공개열쇠를 리용하는 목적과 확인절차들에 관하여 부여된 제한을 가리킨다. 다음것들 즉 수자서명, 비거절, 열쇠암호화, 자료암호화, 열쇠들의 증명서들에 대한 CA서명검증, CRL들에 대한 CA서명검증들중의 하나 또는 그 이상을 가리킬수 있다.
- **비밀열쇠사용주기:** 공개열쇠에 상응한 비밀열쇠리용의 주기를 가리킨다. 일반적으로 비밀열쇠는 공개열쇠의 유효성과 다른 주기에서 리용된다. 실례로 수자서명열쇠들에서 비밀열쇠에 서명하기 위한 리용주기는 일반적으로 공개열쇠을 검증하기 위한 리용주기보다 더 짧다.
- **증명서방략들 :** 증명서들이 여러개의 방략들을 적용하는 환경들에서 리용될수 있다. 이 확장은 증명서가 선택적확인자정보와 함께 지원되는것으로 인정되는 방략들을 렬거한다.
- **방략사영:** 다른 CA들에서 발행된 CA들을 위한 증명서들에서만 리용한다. 방략사영은 발행 CA가 그 발행자의 방략들중의 하나 또는 그 이상을 그 대상의 CA령역에서 리용되는 다른 방략과 동등하게 고찰할수 있다는것을 가리킨다.

증명서대상과 발행자속성

이 확장들은 증명서대상 또는 증명서발행자에 대하여 대안형식들에서 대안이름들을 지원하며 그 증명서대상에 대한 부가정보를 담아 그것이 특정한 사람 또는 실체이라는 증명서사용자의 믿음성을 증가시킬수 있다. 실례로 우편주소와 회사에서의 직위 또는 화상과 같은 정보가 요구될수 있다.

이 령역에서 확장마당들은 다음의것을 포함한다.

- **주동체대안이름:** 어떤 형식을 리용하는 하나 또는 그 이상의 대용이름들을 포함한다. 이 마당은 전자우편, EDI 및 IPSec와 같은 응용들을 지원하는데서 중요하며 그것은 그것들의 소유이름형식들을 리용할수 있다.
- **발행자대안이름:** 여러가지 형식들중에서 어느 하나를 리용하는 하나 또는 그 이상의 대안이름들을 포함한다.
- **주동체등록부속성:** 이 증명서의 대상에 대한 임의의 요구되는 X.500등록부속성을 담는다.

증명경로제약

이 확장들에 의해서 제약명세서(constraint specification)들이 다른 CA들이 그 CA용으로 발행한 증명서들에 포함되도록 한다. 제약들은 대상 CA에 의해 발행될수 있거나 증명련쇄에서 련속 생길수 있는 증명서들의 형들을 제한할수 있다.

이 령역에서 확장마당들은 다음의것을 포함한다.

- **기초제약들(Basic constraints)**: 대상이 CA로 작용할수 있는가를 가리킨다. 그렇다면 증명경로길이제약을 명시할수 있다.
- **이름제약들(Name constraints)**: 증명경로에서 련속한 증명서들에 있는 모든 대상이름들이 위치할 이름공간을 지적한다.
- **방략제약들(Policy constraints)**: 명시적인 방략확인을 요구하거나 또는 증명경로의 나머지부분에 대하여 방략작성을 금지할수 있는 제약들을 명기한다.

참고문헌

BRYA88. Bryant. W. *Designing an Authentication System: A Dialogue in Four Scenes*. Project Athena document, February 1988. Available at http:// web.mit.edu/Kerberos/www/dialogue.html

KOHL94 Kohl, J.; Neuman, B.; and Ts'o, T. "The Evolution of the Kerberos Authentication Service." In Brazier, F., and johansen, D. *Distributed Open Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1994. Available at http://web.mit.edu/Kerberos/www/papers.html.

참고할 web싸이트

- **MIT Kerberos 싸이트**: FAQ, 론문들과 문서들 및 상품싸이트에 대한 지적자들을 포함하는 Kerberos에 대한 정보
- **USC/ISI Kerberos 페지**: Kerberos자료의 다른 자료원천
- **공개열쇠하부구조작업그룹(Public-key Infrastructure Working Group)**: X.509v3에 기초한 IETF그루빠개발규격
- **판본표시(Verisign)**: X.09관련제품들의 기본부동산; 이 싸이트에서 백지들과 기타 가치 있는 자료

문 제

1. PCBC방식(그림 11-7)에서 암호문의 한개 블로크에서의 우연오유가 평문의 모든 련속한 블로크들에로 확산된다는것을 밝히시오.
2. PCBC방식에서 블로크 C_i와 C_{i+1}이 전송중에 교환된다는것을 밝히시오. 이것은 복호된 블로크 P_i와 P_{i+1}에만 영향을 미치지 련속된 블로크들에는 영향을 미치지 않는다는것을 밝히시오.
3. 그림 11-5의 ㄷ에서 설명된 X.509에 대한 원래의 세방향인증절차는 보안결함을 포함한다. 그 규약의 본질은 다음과 같다.

A→B:　　A{t_A , r_A , B}
B→A:　　B{t_B , r_B , A, r_A }
A→B:　　A{r_B}

X.509의 본문은 검사시간도장 t_A와 t_B가 세방향인증에 대한 선택항목이라는것을 설명한다. 다음의 실례를 고찰하자. A와 B가 이전에 진행한 규약을 리용하고 적 C가 선행한 세개의 통보문들을 가로챈다고 하자. 또한 시간도장들이 리용되지 않고 모두 0으로 설정된다고 가정한다. 마지막으로 C는 A가 B로 나타나도록 하려고 한다고 하자. C는 처음 로획한 통보문을 B에게 보낸다.

C→B: A{0, r_A , B}

B는 A에게 보낸다고 생각하고 응답하지만 실제로 C에게 보낸다.

B→C: B{0, r'_B, A, r_A}

C는 그사이에 어떤 방법으로 A가 C에 대한 인증을 하도록 한다. 결과 A는 C에게 다음의것을 보낸다.

A→C: A{0, r'_A ,B}

C는 B가 C에게 제공하는 같은 한번쓰기정보를 리용하여 A에게 응답한다.

C→A: C{0, r'_B, A, r'_A }

A는

A→C: A{ r'_B}

로 응답한다. C는 분명 B가 A와 대화하고 있다는것을 확인시킬 필요가 있을것이며 따라서 받은 통보문을 다시 B에게 돌려 보낸다.

C→B: A{ r'_B}

따라서 B는 사실은 C에게 전달하고 있는것을 A에게 전달한다고 믿게 된다. 시간도장의 리용에 의거하지 않는 이 문제의 간단한 해결책을 제기하시오.

4. X.509의 1988판은 RSA열쇠들이 큰수의 인수분해의 어려움성에 대한 현재의 정보가 주어 지면 안전하여야 한다는 성질들을 들고 있다. 론의는 공개지수와 mod n에 관한 제약으로 결속된다.
평문을 풀기 위하여 mod n의 e차뿌리를 취하며 공격을 막기 위해서는 $e>\log_2(n)$이 성립해야 한다.
그 제약이 정확하다고 해도 그것을 요구하는 리유는 정확치 않다. 주어 진 리유에서 무엇이 잘못이고 무엇이 정확한가?

부록 11: KERBEROS암호화기술

kerberos는 여러가지 암호관련조작들을 지원하는 암호서고를 포함한다.

통과암호-열쇠전송

Kerberos에서 통과암호들은 7-bit ASCII형식으로 표시할수 있는 문자들의 리용으로 국한된다. 이 통과암호는 kerberos자료기지에 기억되여 있는 암호열쇠로 전환된다. 그림 11-6은 그 절차를 보여 준다.

ㄱ) 통과암호를 비트흐름으로 변환

ㄴ) 비트흐름은 입력열쇠로 변환

ㄷ) 통과암호의 DES CBC 검사합의 생성

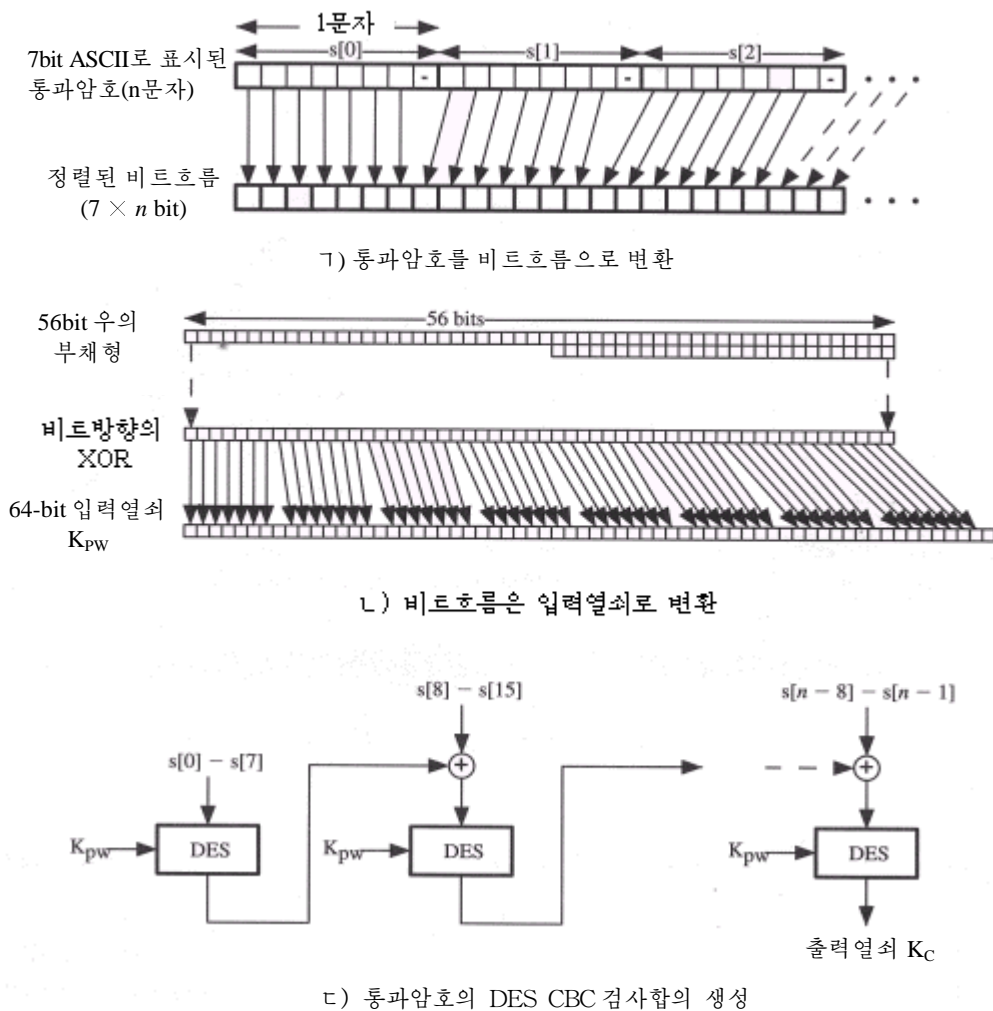



그림 11-6. 통과암호로부터 암호열쇠의 생성

먼저 문자렬 S는 비트렬 b로 묶어 져 첫 문자는 첫 7bit에, 둘째 문자는 두번째 7bit에 등으로 기억된다. 이것은 다음과 같이 표시할수 있다.

b[0] = bit 0 of s[0]
…
b[0] = bit 6 of s[0]
b[7] = bit 0 of s[1]
…
b[7i+m] = bit m of s[I] 0≦ m ≦ 6

다음 그 비트렬은 "부채살"방식으로 비트들을 직선으로 늘여 놓고 비트끼리의 XOR를 수행하여 56bit로 밀집된다. 실례로 그 비트렬의 길이가 59이면

$$b[55]=b[55]\oplus b[56]$$
$$b[54]=b[54]\oplus b[57]$$
$$b[53]=b[53]\oplus b[58]$$

이다.

이것은 56-bit DES열쇠를 생성한다. 확장된 64-bit열쇠형식을 구성하기 위하여 그 렬은 8개의 7-bit블로크들의 렬로 취급되며 입력열쇠 Kpw를 생성하기 위하여 8개의 8-bit블로크들로 넘어 간다.

마지막으로 원래의 통과암호는 DES의 암호블로크련쇄(CBC)방식에서 열쇠 Kpw로 암호화된다. CBC검사합이라고 하는 이 처리로부터 돌려 받은 마지막 64-bit블로크는 이 통과암호와 련관된 출력열쇠이다.
전체 알고리듬은 임의의 통과암호를 64-bit하쉬코드로 넘기는 하쉬함수로 볼수 있다.

확산형암호블로크련쇄방식

DES의 CBC방식의 매 단계에서 DES알고리듬의 입력은 현재의 평문블로크와 바로 앞의 암호블로크의 XOR로 이루어 진다(그림 3-12). 매개 평문블로크들이 독립적으로 암호화되는 전자부호책방식에 비해 이 방식의 우점은 같은 평문블로크가 반복되면 서로 다른 암호문블로크들이 생성되는것이다.

CBC에서는 오유가 암호문블로크 C_I의 전송중에 생기면 그 오유가 회복되는 평문블로크 P_I와 P_{I+1}들에 확산되는 특성이 있다.

Kerberos판본 4는 확산CBC방식(PCBC)이라고 부르는 CBC의 확장을 리용한다 [MEVE82]. 이 방식은 한개의 암호문블로크의 오유가 통보문의 모든 련속한 복호된 블로크들에 확산되는 성질이 있다. 따라서 자료암호와 완정성이 한개의 조작으로 결합된다.

그림 11-7에 PCBC를 보여 주었다. 이 방식에서 암호화알고리듬의 입력은 현재의 평문블로크, 앞의 암호문블로크와 평문블로크의 XOR이다.

$$C_n = E_k[C_{n-1} \oplus P_{n-1} \oplus P_n]$$

복호에서 매개 암호문블로크는 복호알고리듬을 통과한다. 그러면 출력은 선행한 암호문블로크 및 선행한 평문블로크와 XOR된다. 이 방식은 다음과 같이 동작한다.

$$D_K[C_n] = d_k[E_k[C_{n-1} \oplus P_{n-1} \oplus P_n]]$$
$$= C_{n-1} \oplus P_{n-1} \oplus P_n$$
$$C_{n-1} \oplus P_{n-1} \oplus D_k[C_n] = P_n$$

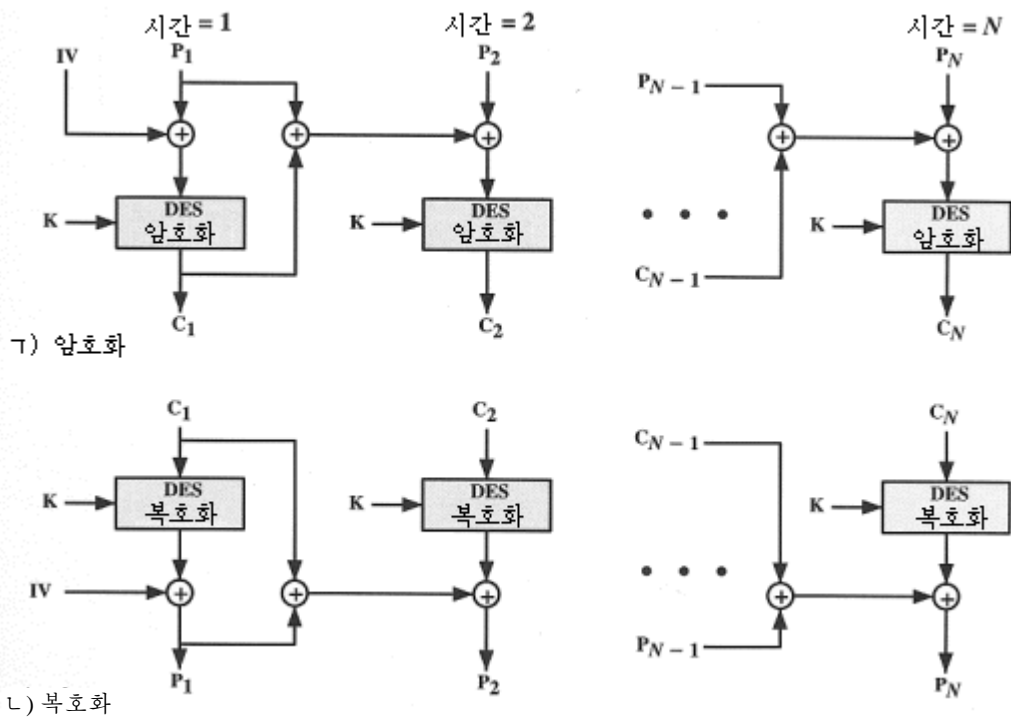



그림 11-7. 확산형암호블로크련쇄(PCBC)방식

제12장. 전자우편보안

전자우편은 거의 모든 분산된 환경들에서 제일 많이 리용되는 망응용이다.

또한 모든 구성방식들과 판매자가동환경전반에서 널리 쓰이는 유일한 분산응용이다. 사용자들은 주콤퓨터의 조작체계나 통신수단에는 관계없이 인터네트에 직접 또는 간접적으로 접속된 다른 사람들에게 우편을 보내게 되기를 바라거나 그렇게 하고 있다.

다양한 목적들을 실현하는데서 전자우편에 대한 의뢰가 폭발적으로 증대하는것과 관련하여 인증과 기밀성에 대한 요구도 높아 지고 있다. 두가지 방식 즉 PGP와 S/MIME가 앞으로 몇년안에 널리 쓰이게 될것으로 예견된다.

12.1 PGP

PGP(PRETTY GOOD PRIVACY)의 개발은 특기할 사변이다. 필 짐머맨(phil zimmermann)이 거의 단독으로 개발한 PGP는 전자우편과 파일보관에 쓰이는 기밀성과 인증봉사를 제공한다. 요컨대 짐머맨(**Zimmermann**)은 다음의 과제들을 실현하였다.

1. 기초블로크로서 가장 실용적인 암호알고리듬을 선택하였다.
2. 이 알고리듬들을 조작체계나 처리기와는 독립이고 쉽게 리용할수 있는 몇개의 명령들에 기초한 상용응용프로그람에 통합하였다.
3. 원천부호를 비롯하여 프로그람과 그의 문서화를 인터네트, 전자게시판 및 콤퓨터봉사와 같은 상업망들에서 자유로 리용할수 있게 하였다.
4. 회사들에 완전호환성의 저가격상업판본의 PGP를 제공하는데 동의하였다.

PGP는 급격히 발전하면서 현재 널리 리용되고 있다. 많은 원인들이 그 성장의 추동력으로 되였다.

1. DOS/Windows, UNIX, Macintosh 등을 비롯하여 여러 가동환경들에서 기동하는 판본들은 세계적으로 무료로 쓸수 있다. 또한 상업판은 판매자가 생산하는 제품들에 대한 사용자의 수요를 충족시킨다.
2. 광범한 공개조사에서도 생명력이 있고 매우 안전하다고 인정되는 알고리듬에 기초한것이다. 특히 RSA, DSS 및 Diffie-Hellman을 공개열쇠암호로서 포함한다. 또한 CAST-128, IDEA 및 3중DES를 전통암호로, SHA-1을 하쉬부호화로서 포함한다.
3. PGP는 파일들과 통보문들을 암호화하는 규격화된 방식을 선택하고 강화하려고 하는 회사로부터 인터네트나 다른 망들을 리용하여 세계적범위에서 다른 사람들과 안전하게 통신하려는 개별적사람들에 이르기까지 광범한 응용범위를 가진다.
4. PGP는 어떤 정부나 규격화기구에 의하여 개발되지도 않았고 또 그것들에 의하여 조종되지도 않는다. 《재정》에 대하여 본능적으로 의혹을 품는 사람들에게는 이것으로 해서 PGP가 아주 흥미 있다.

PGP의 운영에 대한 개괄로부터 시작하여 암호열쇠들을 창조하고 보관하는 방법을 본다. 다음에는 공개열쇠관리의 중요한 문제에 대하여 고찰한다.

기호표식

이 장에서 쓰이는 대부분 기호들은 일부를 제외하고 이전과 같다.

Ks = 전통암호방식에서 쓰이는 대화열쇠
KR_a = 공개열쇠암호방식에서 쓰이는 사용자 A의 비밀열쇠
KU_a = 공개열쇠암호방식에서 쓰이는 사용자 A의 공개열쇠
EP = 공개열쇠암호화함수
DP = 공개열쇠복호화함수
EC = 비밀열쇠암호화함수
DC = 비밀열쇠복호화함수
H = 하쉬함수
|| = 련결
Z = ZIP알고리듬을 리용한 압축
R64 = radix64 ASCII형식에로의 변환

PGP문서화의 공개열쇠암호방식에서 공개열쇠와 쌍을 이루는 열쇠로서 비밀열쇠라는 말을 자주 쓴다. 앞에서 언급한바와 같이 이것은 전통암호에서 쓰던 비밀열쇠와 혼돈을 일으킬수 있다. 따라서 비밀열쇠(private key)라는 용어를 대용한다.

조작해설

PGP의 실지조작은 열쇠관리와 대립되는것으로서 5가지 봉사들 즉 인증, 기밀성, 압축, 전자우편의 호환성 및 토막화(표 12-1)로 구성된다. 그 매개를 차례로 본다.

표 12-1. PGP봉사의 개괄

기능	리용되는 알고리듬들	해설
수자서명	DSS/SHA 또는 RSA/SHA	통보문의 하쉬부호는 SHA-1을 써서 만든다. 이 통보문의 요약정보는 송신자의 비밀열쇠로 DSS나 RSA를 써서 암호화하며 그 통보문에 포함된다.
통보문 암호화	Diffie-Hellman이나 RSA를 리용한 CAST, IDEA 또는 3열쇠3중DES	통보문은 송신자가 생성한 림시 대화열쇠로 CAST-128이나 IDEA 또는 3중DES를 리용하여 암호화한다. 그 대화열쇠는 수신자의 공개열쇠로 Diffie-Hellman이나 KSA를 써서 암호화되며 그 통보문에 포함된다.
압축	ZIP	통보문은 보관이나 전송을 위해 ZIP로 압축할수 있다.
전자우편의 호환성	Radix64-변환	전자우편응용에 투명성을 제공하기 위해 암호화된 통보문은 radix-64변환을 리용하여 ASCII문자렬로 변환할수 있다.
토막화	------	PGP는 최대통보문크기제한을 위하여 토막화와 재조립을 한다.

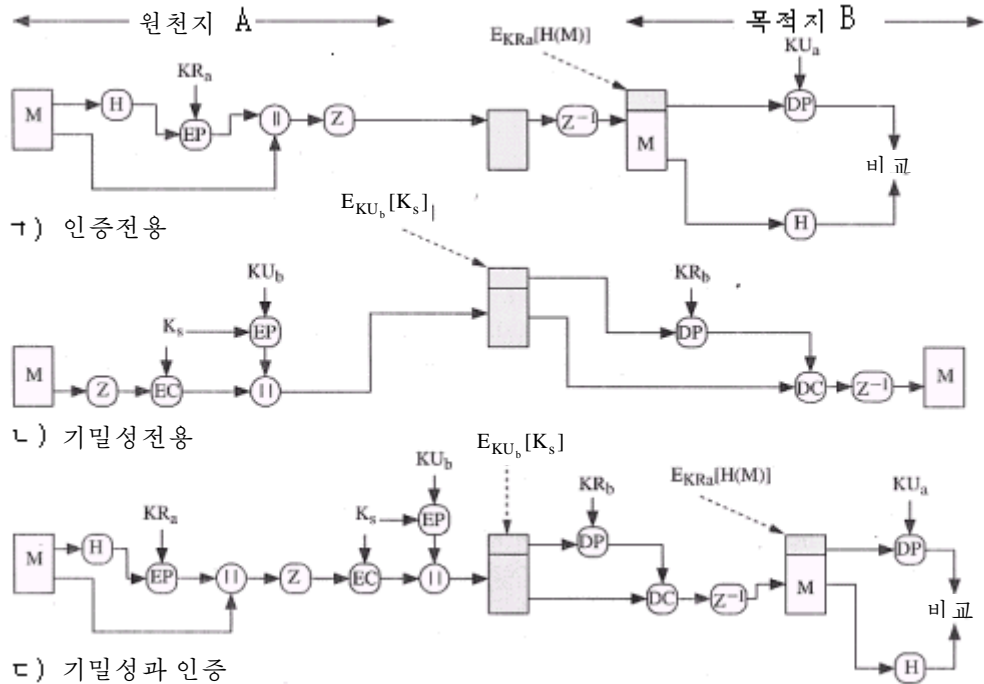



그림 12-1. PGP암호기능

인증

그림 12-1의 ㄱ에 PGP가 제공하는 수자서명봉사를 보여 주었다. 이것이 10장에서 고찰되고 그림 8-5의 ㄷ에서 보여 준 수자서명방식이다. 순서는 다음과 같다.

1. 송신자는 통보문을 작성한다.
2. SHA-1을 리용하여 그 통보문의 160-bit하쉬부호를 생성한다.
3. 그 하쉬부호는 송신자의 비밀열쇠에 의해 RSA로 암호화되며 그 결과는 통보문에 부가된다.
4. 수신자는 RSA를 리용하여 송신자의 공개열쇠로 복호하며 하쉬부호를 재현한다.
5. 수신자는 그 통보문에 대한 새로운 하쉬부호를 생성하고 그것을 복호한 하쉬부호와 비교한다. 만일 그 두개가 일치하면 그 통보문은 정당한것으로 수락된다.

SHA-1과 RSA를 결합하면 효과적인 수자서명방식이 제공된다. RSA강도로부터 수신자는 정확한 비밀열쇠를 가진 사람만이 그 서명을 생성할수 있다는것을 확신한다. SHA-1의 강도로부터 수신자는 그 하쉬부호로부터 기초통보문의 서명과 일치하는 새로운 통보문을 더는 그 누구도 생성하지 못한다고 확신한다.

대안으로서 또한 서명은 DSS/SHA-1을 리용하여 생성할수 있다.

서명이 통보문과 파일에 붙어 있는것이 일반적이나 항상 그렇지는 않다. 즉 붙어 있지 않는 서명들이 지원될 때도 있다. 비부가서명은 서명한 통보문과 따로 보관되며 전

송될수 있다. 이것은 여러가지 원인으로 하여 필요한 때가 있다. 어떤 사용자는 보내거나 받는 모든 통보문들의 개별적서명등록을 하려고 할수 있다. 실행형프로그람의 비부가서명(detached signature)은 이후의 비루스감염을 적발할수 있다. 마지막으로 비부가서명은 한개이상의 대상들이 법적계약과 같은 문서에 서명할 때 리용할수 있다. 매 사람들의 서명은 독립이며 따라서 그 문서에만 적용된다. 그렇지 않으면 서명은 문서와 첫번째 서명에 서명한 두번째 서명자와 다음 서명자들에 의해 차례로 중복된다.

기밀성

PGP가 제공하는 다른 기본적인 봉사는 기밀성이다. 그것은 통보문들을 암호화하여 전송하거나 파일로 국부적으로 보관하는데 리용한다. 두 경우 다 전통암호알고리듬 CAST-128을 리용한다.그렇지 않으면 IDEA나 3중DES를 리용할수도 있다. 64-bit암호반결합(CFB)방식도 리용된다.

여기서 열쇠배포문제가 제기된다. PGP에서는 매 비밀열쇠를 한번씩만 리용한다. 즉 매개 통보문에 대해 우연128-bit수로서 새 열쇠가 생성된다. 따라서 이것은 문서화에서 열쇠로 되는데 그것은 사실상 1회용열쇠이다. 그것이 한번만 쓰이는것이므로 대화열쇠는 그 통보문에 포함되여 그것과 함께 전송된다. 그 열쇠를 보호하기 위하여 수신자의 공개열쇠로 암호화한다.

그림 12-1의 ㄴ에 보여 준 그 절차는 다음과 같다.

1. 송신자는 통보문과 우연128-bit수를 생성하고
2. CAST-128(또는 IDEA 혹은 3중DES)을 리용하여 통보문을 암호화한다.
3. 대화열쇠는 수신자의 공개열쇠에 의하여 RSA로 암호화되여 통보문에 첨부된다.
4. 수신자는 자기의 비밀열쇠로 복호하여 대화열쇠를 회복한다.
5. 그 대화열쇠는 그 통보문을 복호하는데 리용된다.

열쇠암호화를 위하여 RSA를 사용하는데 대한 대안으로서 PGP는 디피-헬만(Diffie-Hellman)이라고 하는 선택항목을 제공한다. 6장에서 설명한것처럼 Diffie-Hellman은 열쇠교환알고리듬이다. 사실 PGP는 EIGamal(문제 6.19를 볼것)로 알려 진 암호화/복호화를 제공하는 Diffie-Hellman의 변종을 리용한다.

몇가지 조사를 진행하였다. 첫째로, 암호처리시간을 줄이기 위하여 전통암호와 공개열쇠암호의 결합을 리용한다. 둘째로, 수신자만이 통보문에 국한되여 있는 대화열쇠를 발견할수 있으므로 공개열쇠알고리듬에 의해 대화열쇠배포문제를 해결할수 있다 .진행중의 대화는 아직 이야기되지 않았기때문에 6장에서 론의된 형태의 대화열쇠교환규약이 필요하지 않다.

매개 통보문들은 그것의 열쇠와 일시독립인 사건이다. 더우기 전자우편의 보관 및 전송의 성질이 있으므로 두가지 측면이 같은 대화열쇠를 가진다는것을 인증하는데 접어 드는것은 현실적이 못된다. 마지막으로 1회용상용열쇠의 리용이 강한 전통암호방법이라는 것을 강조해 둔다. 통보문의 극히 일부만이 매개 열쇠로 암호화되는데 그 열쇠들사이에는 관련이 없다. 이리하여 공개열쇠알고리듬이 안전한만큼 전체 방식도 안전하다. 끝으로 PGP는 사용자에게 열쇠크기선택을 768bit로부터 3072bit까지의 범위에서 제공한다(서명용DSS열쇠는 1024bit까지로 제한된다).

기밀성과 인증

그림 12-1의 ㄷ에서 보여 준것처럼 두 형태의 봉사가 한 통보문에 쓰일수 있다. 먼저 서명이 평문통보문에 대하여 생성되고 그 통보문에 첨부된다. 다음 서명된 그 평문통보문은 CAST-128(또는 IDEA 혹은 3중DES)로 암호화된다. 대화열쇠는 RSA(또는 ELGamal)로 암호화된다. 그 차례는 반대로 즉 통보문을 암호화하고 다음 암호화된 통보문에 대하여 서명을 생성하는것이 더 적합할 때도 있다. 서명을 통보문의 평문과 함께 보관하는것이 일반적으로 더 편리하다.

더우기 제3자에 대한 검증에서 서명이 먼저 진행되면 제3자는 그 서명을 검증할 때 상용열쇠를 리용할 필요가 없다.

요약하여 말하면 두 봉사가 리용될 때 송신자는 먼저 자기의 비밀열쇠로 통보문에 서명한 다음 대화열쇠로 그 통보문을 암호화한 다음 그 대화열쇠를 수신자의 공개열쇠로 암호화한다.

압축

기정적으로 PGP는 서명을 생성하고 암호화하기전에 통보문을 압축한다. 이것은 전자우편전송과 파일보관에서 공간을 절약하는데 유리하다.

그림 12-1에서 압축을 Z로, 푸는것을 Z^{-1}로 하여 보여 준 압축알고리듬의 배치는 다음의 경우에 위험하다.

1. 서명은 두가지 리유로 압축전에 생성된다.
   ㄱ) 후에 검증을 위하여 압축되지 않은 통보문을 서명과 함께 보관할수 있도록 암호화되지 않은 통보문에 서명하는것이 더 낫다. 만일 압축된 문서에 서명했다면 후날의 검증을 위해 통보문을 압축한 판본을 보관하거나 검증이 요구될 때 그 통보문의 압축을 풀어야 할 필요가 있게 된다.
   ㄴ) 지어 검증을 위해 압축 푼 통보문을 동적으로 생성한다면 PGP의 압축알고리듬은 곤난성을 배출한다. 그 알고리듬은 결정적이 아니다. 즉 알고리듬의 각이한 실현에 의해 실행속도 대 압축비률에서 서로 다르므로 결과 서로 다른 압축형식들을 낳는다. 그러나 이 서로 다른 압축알고리듬은 임의의 판본의 알고리듬이 어떤 다른 판본의 결과를 정확히 압축할수 있으므로 량립가능하다. 하쉬함수를 적용하여 압축한 다음 서명하는것은 같은 압축알고리듬에 대한 모든 PGP개선을 강조한다.
2. 암호보안을 강화하기 위하여 통보문암호화를 압축후에 진행한다. 압축된 통보문은 원래의 평문보다 파잉이 적으므로 암호분석이 더 힘들다.

압축알고리듬으로서 ZIP를 리용하는데 이것을 부록 12-1에 주었다.

전자우편의 호환성

PGP를 리용할 때 적어도 전송할 블로크의 일부는 암호화된다. 만일 서명봉사만 리용한다면 그 통보문의 요약정보가 암호화된다(송신자의 비밀열쇠로).그리고 기밀성봉사만을 리용한다면 서명을 포함한 통보문이 암호화된다(1회용대칭열쇠로). 이리하여 결과 블로크들의 일부 또는 전체는 우연 8-bit옥테드(Octets)로 구성된다.

그러나 많은 전자우편체계들은 ASCII본문으로 이루어 지는 블로크들만을 리용한다. 이 제한을 완화시키기 위해 PGP는 8bit의 2진렬을 인쇄가능한 ASCII문자들의 렬로 보관하는 봉사를 제공한다.

그에 리용되는 방식이 radix-64변환이다. 2진자료의 세 옥테트들중에서 매개는 4개의 ASCII문자들에 대응된다. 이 형식에 역시 전송오유를 검출하기 위해 CRC을 덧단다. 부록 12-2에 자세히 서술되였다.

Radix64를 리용하면 통보문을 33%까지 확장한다. 다행히도 대화열쇠와 그 통보문의 서명부분이 상대적으로 조밀하고 평문통보문은 압축된다. 사실 압축은 radix-64확장을 보장하고도 남는다. 실례로 [HELD96]에서는 ZIP를 리용하여 평균 약 2.0의 압축비률을 얻는다. 비교적 작은 서명과 열쇠부분들을 무시하면 일반적으로 길이가 X인 파일의 압축과 확장의 전반적효과는 $1.33 \times 0.5 \times X = 0.665X$이다. 따라서 총적으로 약 3분의 1이 압축된다.

Radix-64알고리듬에서 한가지 주목할만 한 점은 그것의 입력이 ASCII본문이라고 해도 입력흐름을 내용에 관계없이 radix-64형식으로 맹목적으로 변환한다는것이다. 따라서 통보문이 서명은 되고 암호화되지 않은채로 변환이 전체 블로크에 적용되면 그 출력은 다른 사람은 읽을수 없는데 이것은 일정한 수준의 기밀성을 제공한다. 또한 PGP는 서명된 평문통보문의 서명부분만을 radix-64형식으로 변환할수 있도록 구성되였다. 이것은 PGP를 쓰지 않고 수신자가 그 통보문을 읽을수 있게 한다. 그러나 여전히 PGP는 서명을 검증하는데 리용할수 있다.

그림 12-2에 지금까지 론의된 4가지 봉사들사이의 호상관계를 보여 주었다. 전송에서 서명이 요구되면 그것을 압축된 평문의 하쉬부호를 리용하여 생성한다. 그때 평문과만일 있다면 서명까지 합치여 압축한다. 다음 기밀성이 요구되면 블로크(압축된 평문이나 서명에 평문을 합치여 압축한것)를 암호화하고 공개열쇠로 암호화된 상용암호열쇠를 첨부

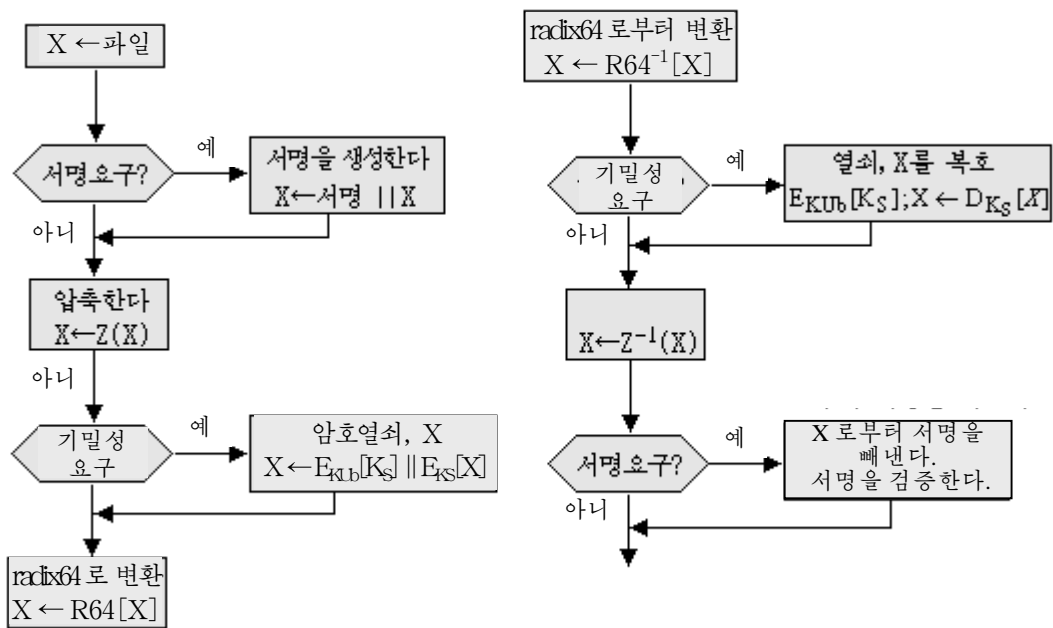



그림 12-2. PGP통보문들의 전송과 수신

한다. 마지막으로 전체 블로크를 radix-64형식으로 변환한다.

수신측에서는 들어 온 블로크를 먼저 radix-64형식으로 다시 변환된다.

다음 그 통보문이 암호화되였으면 수신자는 대화열쇠를 회복하고 통보문을 복호한다. 그리고 결과블로크의 압축을 풀기한다. 그 통보문이 서명되였으면 수신자는 전송된 하쉬부호를 회복하여 그것을 자기가 계산한 하쉬부호와 비교한다.

토막화와 재조립

전자우편수단들은 최대통보문길이를 제한한다. 실례로 인터네트에서 접근할수 있는 수단들에는 50000옥테드의 최대길이가 부여된다. 이것보다 더 긴 통보문들은 토막들로 분할되여 그 매개를 따로따로 전송한다.

이 제한을 도모하기 위하여 PGP는 자동적으로 너무 큰 통보문을 전자우편으로 보낼수 있도록 충분히 작은 토막들로 나눈다. 토막화는 radix-64변환을 포함하여 다른 모든 처리들이 끝난 다음에 진행된다. 따라서 대화열쇠요소와 서명요소는 첫 토막의 시작에 한번씩만 나타난다. 수신마지막에 PGP는 전자우편머리부들을 모두 해제하여 그림 12-2의 ㄴ에서 설명한 단계들을 실행하기전에 전체 초기블로크들을 재조립해야 한다.

암호열쇠와 열쇠고리

PGP는 4가지 형태의 열쇠들 즉 1회대화용상용열쇠, 공개열쇠, 비밀열쇠 및 통과단계형식의 비밀열쇠들이다. 이 열쇠들에 대하여 세개의 독립적인 요구조건들이 나온다.

1. 예측 불가능한 대화열쇠를 생성하는 방법이 필요하다.
2. 한명의 사용자가 다중공개열쇠/비밀열쇠쌍을 가진다고 하자. 그 리유는 사용자가 자기의 열쇠쌍을 때에 따라서 바꾸려고 하기때문이다. 그렇게 되면 관흐름에서 임의의 통보문은 페멸된 열쇠로 만들어 진다. 더우기 수신자들은 갱신이 그들에게 알려 질 때까지 낡은 공개열쇠밖에 모른다. 열쇠들을 시간에 따라 변화시킬 필요외에 사용자는 주어 진 시간에 서로 다른 기자그룹들과 대화를 진행하거나 간단히 임의의 한 열쇠로 암호화된 자료량을 제한하여 보안을 강화하기 위하여 다중열쇠쌍들을 가지려고 한다. 이 모든것으로부터 사용자들과 그들의 공개열쇠들사이에서 1대1대응이 성립되지 않는다. 따라서 개별적인 열쇠들을 식별하는 어떤 방법이 필요하다.
3. 매 PGP실체는 기자들의 공개열쇠파일과 마찬가지로 자기의 공개열쇠/비밀열쇠쌍들의 파일을 보관해야 한다.

이제 그 요구들을 차례로 보자.

대화열쇠생성

매 대화열쇠는 하나의 통보문과 관련되며 그 통보문을 암호화 및 복호하기 위하여서만 리용된다. 통보문암호화/복호는 대칭암호알고리듬으로 진행한다. CAST-128과 IDEA는 128-bit열쇠를 리용하며 3중DES는 168-bit열쇠를 리용한다. 다음의 론의는 CAST-128을 가정하고 진행한다.

우연128bit의 수들은 CAST-128자체를 리용하여 생성한다. 우연수발생기의 입력은 128-bit열쇠와 암호화할 평문으로 취급되는 두개의 64-bit암호문블로크들을 만드는데 그것은 128-bit대화열쇠를 만드는데 련결된다. 리용되는 알고리듬은 ANSI X12.17에 기초한다.

두개의 64-bit블로크들로 구성되는 우연수발생기의 "평문"입력은 128bit의 우연수들의 렬로부터 유도된다. 이 수들은 사용자의 건누름에 기초한다. 건누름시간측정과 실제의 건눌림은 둘다 우연렬을 생성하는데 리용된다. 따라서 만일 사용자가 임의의 열쇠를 자기의 보통속도에 맞추면 합리적으로 "우연"인 입력이 생성된다. 또한 이 우연입력은 CAST-128의 전단계 대화열쇠출력과 결합하여 발생기의 열쇠입력을 창조한다. CAST-128의 효과적인 짜임이 주어 지면 그 결과는 예측 불가능한 대화열쇠들의 렬을 생성한다.

부록 12-3에 PGP우연수생성기술이 보다 상세히 소개되였다.

열쇠식별자

암호화된 통보문은 리용된 대화열쇠의 암호화된 형식을 동반한다. 대화열쇠 그자체는 수신자의 공개열쇠로 암호화된다.

따라서 정당한 수신자만이 대화열쇠를 회복할수 있으며 나아가서 통보문을 복호할수 있다. 만일 매 사용자가 하나의 공개/비밀열쇠쌍을 리용한다면 수신자는 자기의 유일한 비밀열쇠인 대화열쇠를 복호하는데 어느 열쇠를 사용하겠는가를 다른 정보가 없이도 결정한다. 그러나 임의의 주어 진 사용자가 다중공개열쇠/비밀열쇠쌍들을 가질수 있다고 하자.

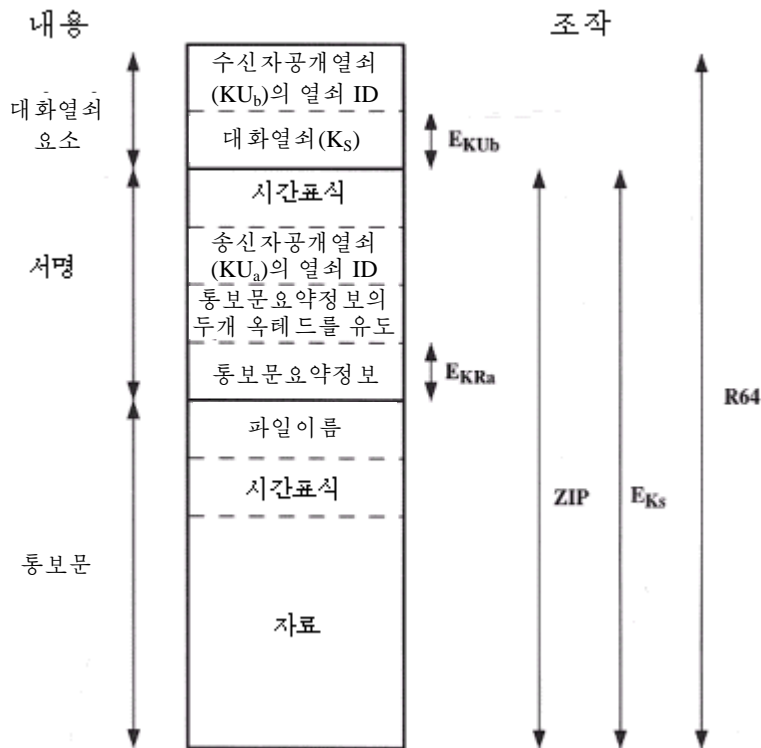
그러면 수신자는 대화열쇠를 암호화하는데 어느 공개열쇠가 리용되였는가를 어떻게 알겠는가?

한가지 방법은 공개열쇠를 통보문과 함께 전송하는것이다. 그러면 수신자는 이것이 정말 그 공개열쇠인가를 검증하고 복호를 진행할수 있다. 이 방식은 동작할수는 있지만 공간랑비가 불필요하게 많다. RSA공개열쇠는 수십자리의 십진수이다. 다른 방도는 기껏 한명의 사용자에게 유일한 매 공개열쇠들에 식별자를 관련시키는것이다. 즉 사용자 ID와 열쇠ID의 결합은 열쇠를 유일하게 확인하는데 충분하다. 그러면 훨씬 더 짧은 열쇠 ID만을 전송하면 된다. 그러나 이 방도는 관리 및 부가금문제를 일으킨다. 즉 열쇠 ID들은 송신자와 수신자가 둘다 열쇠ID로부터 공개열쇠를 얻을수 있도록 보관되여야 한다. 이것은 불필요한 부담일수 있다.

PGP에 대한 해결방도는 매개 공개열쇠에 열쇠ID를 할당하는것이다. 매개 공개열쇠와 관련한 열쇠ID는 그것의 최소의미 있는 64bit로 이루어 진다. 즉 공개열쇠 KUa의 열쇠ID는 (KUa mod 2^{64})이다. 이것은 열쇠가 반복될 확률이 매우 작은 충분한 길이이다.

또한 열쇠ID는 PGP수자서명에도 요구된다. 송신자가 여러개의 비밀열쇠들중의 하나를 리용하여 통보문요약정보를 암호화하므로 수신측은 어느 열쇠가 쓰이였는가를 알아야 한다. 따라서 통보문의 수자서명요소는 대응하는 공개열쇠의 열쇠ID를 포함한다. 그 통보문을 수신하면 수신측은 그 열쇠ID가 해당 송신자에게서 받은 공개열쇠라는것을 검증한 다음에 서명을 검증한다.

열쇠ID의 개념을 도입하였으므로 통보문전송형식을 더 자세히 볼수 있다(그림 12-3). 통보문은 세 요소 즉 통보문요소, 서명(선택적) 및 대화열쇠요소(선택적)들로 이루어 진다.

통보문요소는 파일이름이나 창조시간을 서술한 시간도장이나 보관 또는 전송될 실지의 자료를 포함한다.

기호표시

E_{KUb} = 사용자 b의 비밀열쇠로 암호화

E_{KUa} = 사용자 a의 공개열쇠로 암호화

E_{K_S} = 대화열쇠로 암호화

ZIP = Zip압축함수

R64 = Radix-64변환함수

그림 12-3. PGP통보문의 일반적인 형식

서명요소는 다음의 요소들을 포함한다.

- **시간도장:** 그 서명이 생성된 시간
- **통보문요약정보:** 송신자의 전용서명열쇠로 암호화된 160-bit SHA-1요약정보. 요약정보는 통보문요소의 자료부분에 련결된 서명시간도장에 대하여 계산된다. 요약정보에 서명시간도장을 포함시키는것은 재연(반복)들에 대처하기 위해서이다. 통보문요소의 파일이름과 시간도장부분들을 뺀것은 부가하지 않은 서명이 그 통보문의 머리부에 부가된 서명과 꼭 같다는것을 담보한다. 부가되지 않은 서명들은 아무런 통보문요소의 머리부파일들을 가지지 않는 개별적파일들에 대하여 계산된다.
- **통보문요약정보의 두 선두옥테드:** 수신측이 정확한 공개열쇠가 통보문요약정보를 복호하는데 쓰이였는가를 결정할수 있게 하기 위해 첫 두개의 옥테드의 평

문복사를 복호할 요약정보의 첫 두개의 옥테드와 비교한다. 이 옥테드들은 통보문에 대한 16-bit프레임-검사렬로도 리용된다.

- **송신자공개열쇠의 열쇠ID:** 통보문요약정보를 복호하는데 쓰이는 공개열쇠를 식별하고 여기로부터 그 통보문요약정보를 암호하는데 쓰였던 비밀열쇠을 식별한다.

통보문요소와 선택적서명요소는 ZIP로 압축되며 대화열쇠로 암호화된다.

대화열쇠요소는 대화열쇠와 그것을 암호화하는데 송신자가 리용한 수신측의 공개열쇠의 식별자를 포함한다. 전체 블로크들은 보통 radix-64부호로 암호화된다.

열쇠고리

앞에서 열쇠ID들이 PGP조작에 어떻게 위험하며 기밀성과 인증을 제공하는 임의의 PGP통보문에 두개의 열쇠ID가 어떻게 포함되는가를 보았다. 모든 대방들은 효률적이고 효과적인 리용을 위하여 체계적인 방법으로 이 열쇠들을 보관 및 조직화할 필요가 있다. PGP에서 리용되는 방식은 매 마디에 한쌍의 자료구조를 제공하는데 하나는 그 마디에 부여된 공개열쇠/비밀열쇠쌍을 보관하는 자체구조이고 다른것은 그 마디에 알려 진 다른 사용자들의 공개열쇠를 보관하는 자료구조이다. 이 자료구조들을 각각 비밀열쇠고리와 공개열쇠고리라고 부른다.

시간도장	열쇠ID*	공개열쇠	암호화된 비밀열쇠	사용자ID*
⋮	⋮	⋮	⋮	⋮
T_i	$KU_i \bmod 2^{64}$	KU_i	$E_{H(P_i)}[KR_i]$	사용자$_i$
⋮	⋮	⋮	⋮	⋮

비밀열쇠고리

시간도장	열쇠ID*	공개열쇠	소유자신용	사용자ID*	열쇠합법성	서명(S)	서명신용(S)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	$KU_i \bmod 2^{64}$	KU_i	신용_flag$_i$	사용자$_i$	신용_flag$_i$		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

공개열쇠고리

그림 12-4. 비밀열쇠 및 공개열쇠고리의 일반구조

그림 12-4에 비밀열쇠고리의 일반구조를 보여 주었다. 고리를 표화하여 볼수 있는데 거기서 매 렬은 사용자에게 부여된 공개열쇠/비밀열쇠쌍들의 하나를 표현한다. 매 렬에는 다음의것들이 포함된다.

- **시간도장:** 이 열쇠쌍이 생성된 날자/시간
- **열쇠ID:** 공개열쇠의 최소의미 있는 64bit
- **공개열쇠:** 그 쌍의 공개열쇠부분
- **비밀열쇠:** 그 쌍의 비밀열쇠부분으로서 이 마당은 암호화된다.
- **사용자ID:** 일반적으로 이것은 사용자의 전자우편주소(실례로 @ acm.org)이다. 그러나 사용자는 매개 쌍들에 서로 다른 이름을 관련시켜 선택하거나 같은 사용자의ID를 한번이상 재리용할수 있다.

비밀열쇠고리는 사용자ID나 열쇠ID로 지적할수 있다. 후에 두가지 지정방법에 대한 필요성을 본다.

비밀열쇠고리가 열쇠쌍들을 창조하여 소유하고 있는 사용자의 콤퓨터에만 보관되고 그 사용자에게만 접근가능하다고 할지라도 비밀열쇠의 값을 될수록 안전하게 하는것이 의미가 있다. 따라서 비밀열쇠 그자체는 열쇠고리에 보관되지 않는다. 오히려 이 열쇠는 CASR-128(또는 IDEA나 3중DES)에 의해 암호화된다. 그 절차는 다음과 같다.

1. 사용자는 비밀열쇠들을 암호화하는데 리용하는 통과단계를 선택한다.
2. 체계가 RSA를 리용하여 새로운 공개열쇠/비밀열쇠쌍을 생성할 때 체계는 사용자에게 통과단계를 요구한다. SHA-1을 리용하여 160-bit하쉬부호가 통과단계로부터 생성되며 다음 그 통과단계는 제거된다.
3. 체계는 128bit의 하쉬부호를 열쇠로 하는 CAST-128을 써서 비밀열쇠를 암호화한다. 다음 그 하쉬부호는 삭제되고 암호화된 비밀열쇠는 비밀열쇠고리에 보관된다.

따라서 사용자가 비밀열쇠을 찾으려고 비밀열쇠고리에 접근할 때 그는 통과단계를 제공하여야 한다. PGP는 암호화된 비밀열쇠을 회복하고 통과단계의 하쉬부호를 생성하며 CAST-128에 의해 하쉬부호로 써서 암호화한 비밀열쇠고리를 복호한다.

이것은 매우 치밀하며 효과적인 방식이다. 통과암호에 기초한 임의의 체계들에서와 같이 이 체계의 안전성은 그 통과암호의 안전성에 의존한다. 유혹물이 섞여 드는것을 피하기 위하여 사용자는 쉽게 추측할수는 없지만 쉽게 기억할수 있는 통과단계를 사용해야 한다.

그림 12-4에 공개열쇠고리의 일반구조도 보여 주었다. 이 자료구조는 자기가 아는 다른 사용자들의 공개열쇠를 보관하는데 리용된다. 그 표에서 다음의 일부 마당들만을 고찰하자.

- **시간도장:** 이 기입이 생성된 날자/시간
- **열쇠ID:** 이 기입에 대하여 공개열쇠의 최소의미 있는 64bit
- **공개열쇠:** 이 기입에 대한 공개열쇠
- **사용자ID:** 이 열쇠의 소유자. 여러개의 사용자ID들이 한개의 공개열쇠에 관련될수 있다.

공개열쇠고리는 사용자ID와 열쇠ID에 의해 지적할수 있다. 암호화의 두 수법의 필요성에 대하여서는 뒤에서 고찰한다.

이제는 이 열쇠고리들이 통보문전송과 수신에 어떻게 쓰이는가를 밝혀야 한다. 간단히 하기 위하여 다음의 론의들에서 압축과 radix-64변환은 무시한다. 먼저 통보문전송

을 고찰하는데(그림 12-5) 그 통보문을 서명한 다음 암호화한다고 가정한다. 송신하는 PGP실체는 다음의 단계들을 수행한다.

1. 통보문서명
   ㄱ) PGP는 첨수로서 your_userid를 리용하여 비밀열쇠고리로부터 송신자의 비밀열쇠를 회복한다. 만일 your_userid가 지령으로 제공되지 않았다면 그 고리우의 첫 비밀열쇠가 회복된다.
   ㄴ) PGP는 비밀열쇠를 회복하기 위해 사용자에게 통과단계를 재촉한다.
   ㄷ) 통보문의 서명요소가 구성된다.
2. 통보문암호화
   ㄱ) PGP는 대화열쇠를 생성하여 통보문을 암호화한다.
   ㄴ) PGP는 her_userid를 첨수로 리용하여 공개열쇠고리로부터 수신측의 공개열쇠을 찾는다.
   ㄷ) 통보문의 대화열쇠요소가 구성된다.

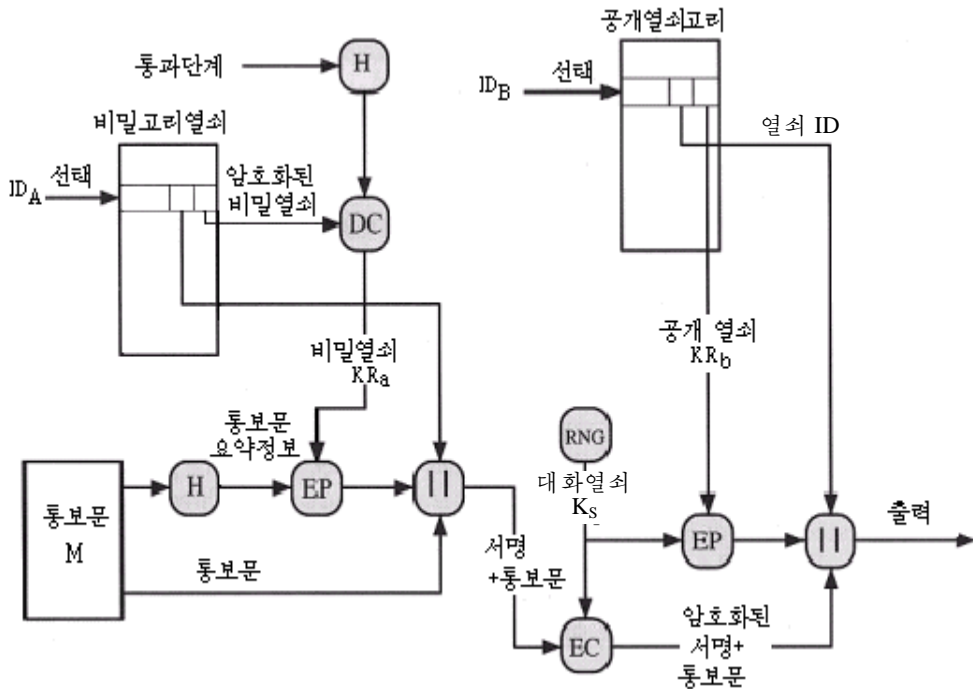



그림 12-5. PGP통보문생성(A→B:압축이나 Radix-64변환은 없다.)

수신PGP실체는 다음의 단계들을 실행한다(그림 12-6).

1. 통보문복호
   1) PGP는 통보문의 대화열쇠요소의 열쇠ID마당을 첨수로 리용하여 비밀열쇠고리로부터 수신자의 비밀열쇠를 회복한다.

2) PGP는 암호화되지 않은 비밀열쇠를 회복하기 위해 사용자에게 통과단계를 재촉한다.
3) 다음 PGP는 대화열쇠를 회복하며 그 통보문을 복호한다.

2. 통보문인증
1) PGP는 통보문의 서명열쇠요소에서 열쇠ID마당을 첨수로 리용하여 공개열쇠고리로부터 송신자의 공개열쇠를 회복한다.
2) PGP는 전송된 통보문요약정보를 회복한다.
3) PGP는 수신된 통보문에 대하여 그 통보문요약정보를 계산하고 그것을 전송된 통보문요약정보와 비교하여 인증한다.

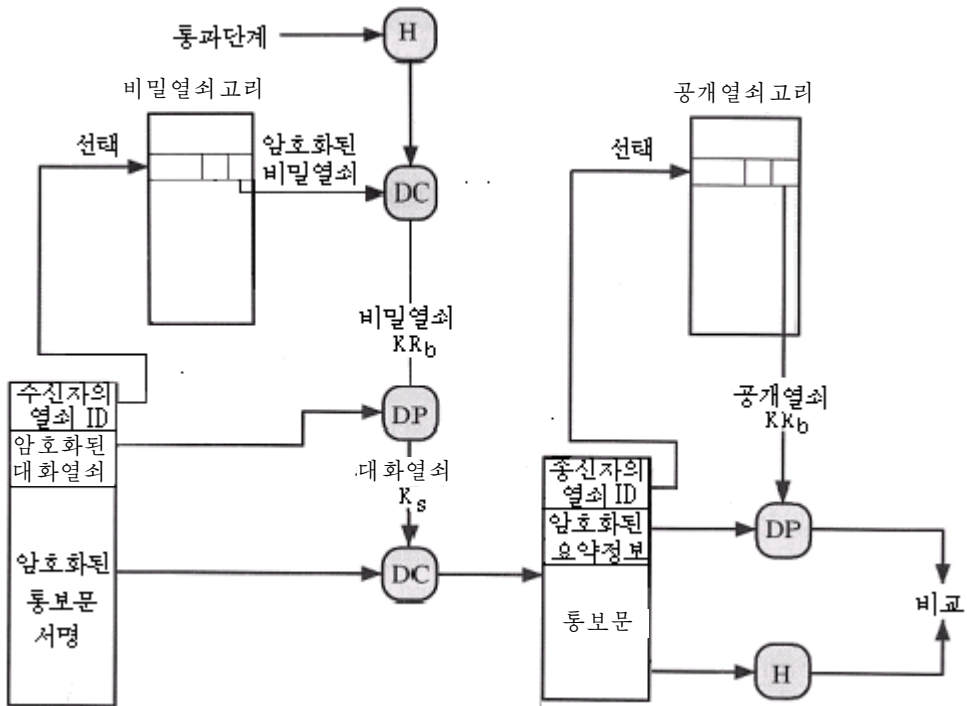



그림 12-6. PGP통보문수신(A→B: 압축이나 radix-64변환은 없다.)

공개열쇠관리

지금까지의 론의로부터 알수 있는것처럼 PGP는 기묘하고 능률적이며 서로 꼭 맞물린 기능들의 모임과 유효한 기밀성과 인증봉사를 제공하는 형식들을 포함한다. 체계를 완결 짓기 위하여 공개열쇠를 관리하는 마지막령역을 소개한다. PGP문서화는 이 령역의 중요성을 놓치지 않는다.

가로채기로부터 공개열쇠들을 보호하는 이 모든 처리들은 실제 공개열쇠응용들에서 가장 힘든 유일한 문제이다. 그것은 공개열쇠암호의 "Achilles heel"이며 많은 쏘프트웨어복잡성은 이 하나의 문제를 푸는데 기인한다.

PGP는 리용할수 있는 몇개의 제안된 선택들에 의하여 이 문제를 푸는 구조를 제공한다. PGP는 여러가지의 형식적 및 비형식적환경들을 리용하도록 되여 있으므로 이 장의 뒤에서 S/MIME의 론의에서 보게 되는것과 같이 확정된 공개열쇠관리방식을 설정할수 없다.

공개열쇠관리에 대한 입문

문제의 본질은 다음과 같다. 사용자 A는 PGP를 리용하여 다른 사람들과 대화하기 위하여 그들의 공개열쇠들을 포함하는 공개열쇠고리를 창조한다. A의 열쇠고리에 B의 공개열쇠가 포함되며 그 열쇠는 사실상 C가 준것이라고 가정한다. 이런 현상은 실례로 A가 공개열쇠를 보내기 위하여 B가 사용한 게시판광고체계(BBS)로부터 그 열쇠를 얻을 때 생긴다. 그 결과에는 두개의 위험이 존재하게 된다. 첫째로, C는 통보문을 A에게 보내여 A가 그 통보문이 B에게서 온것으로 받아 들이도록 B의 서명을 위조할수 있다. 둘째로, A로부터 B로 전송되는 다른 임의의 암호화된 통보문을 C가 읽을수 있다.

사용자의 공개열쇠고리에 거짓공개열쇠가 포함될 위험을 최소화하는 많은 방법들이 있을수 있다. A가 B에 대한 믿을만한 공개열쇠를 얻으려고 한다고 가정한다. 그때의 몇가지 방법들은 다음과 같다.

1. 물리적으로 B로부터 열쇠를 얻는다. B는 플로피디스크에 자기의 공개열쇠 (KU_b)를 기억시키고 그것을 A에게 넘겨 줄수 있다. A는 그 열쇠를 자기의 체계에 그 플로피디스크로부터 태운다. 이것은 매우 안전한 방법이지만 실천적제한들이 있다.
2. 전화로 열쇠를 검증한다. A가 전화로 B를 인식할수 있으면 A는 B를 찾아 그에게 전화로 radix-64형식으로 열쇠를 지적할것을 요구한다. 더 현실적인 방안으로서 B는 자기의 열쇠를 전자우편으로 A에게 보낸다. A는 PGP로 그 열쇠의 160-bitSHA-1요약정보를 만들고 그것을 16진수형식으로 현시할수 있다. 이것을 열쇠의 《지문》이라고 부른다. 다음 A는 B를 호출하고 그에게 전화로 그 지문을 불러 줄것을 요구할수 있다. 만일 그 두개의 지문이 일치하면 열쇠는 검증된다.
3. 호상 신용되는 대상 D로부터 B의 공개열쇠를 얻는다. 이를 위하여 안내자 D는 서명된 증명서를 만든다. 그 증명서는 B의 공개열쇠, 그 열쇠의 창조시간, 그 열쇠의 유효성주기를 포함한다. D는 이 증명서의 SHA-1요약정보를 생성하고 그것을 자기의 비밀열쇠로 암호화하여 그 증명서에 대한 서명에 부친다. D만이 그 서명을 창조할수 있으므로 그밖의 누구도 거짓공개열쇠을 만들어 그것이 D가 서명한것처럼 할수 없다. 서명된 증명서는 직접 B나 D가 A에게 보내거나 전자게시판상에서 전송할수 있다.
4. 신용되는 증명국으로부터 B의 공개열쇠를 얻는다. 다시 공개열쇠증명서가 그 국에 의해 창조되고 서명된다. 다음 A는 국에 사용자이름을 제공하고 서명된 증명서를 받을수 있다.

3과 4의 경우에 대하여 A는 안내자의 공개열쇠의 복사를 가지고 그것이 정당하다는것을 확신하여야 한다. 종국적으로 안내자로 활동하는 사람에게 신용수준을 부여하는것은 A의 책임이다.

신용의 리용

PGP는 증명국이나 신용을 창설하는데 필요한 아무런 명세서도 포함하지 않지만 신용을 리용하고 신용에 공개열쇠들을 련관시키고 신용정보를 탐색하는 편리한 방법들을 제공한다.

기본구조는 다음과 같다. 공개열쇠고리에서 매 입력은 앞의 소절에서 설명한것처럼 공개열쇠증명서이다. 이러한 매 입력들은 PGP가 이것이 그 사용자에게 정당한 공개열쇠이라는것을 믿는 범위를 지적하는 **열쇠합법성마당**에 관련된다. 즉 믿음성수준이 높으면 높을수록 사용자ID는 그 열쇠에 더 강하게 속박된다. 이 마당은 입력의 **서명신용마당**들의 모임으로부터 유도된다. 마지막으로 매개 입력들은 특정의 소유자와 관련한 공개열쇠를 정의하며 이 공개열쇠에는 다른 공개열쇠증명서들에 서명하는데 신용되는 정도를 가리키는 **소유자신용마당**이 포함된다. 즉 이 수준의 신용은 사용자에 의하여 부여된다. 서명신용마당들을 다른 입력으로부터 소유자신용마당의 캐시된 복사라고 생각할수 있다.

앞절에서 언급한 세개의 마당들은 신용기발바이트라고 부르는 구조에 각각 포함된다. 이 세개의 매 리용에 대한 신용기발의 내용을 표 12-2에 보여 주었다.

표 12-2. 신용기발바이트의 내용

공개열쇠소유자에게 부여된 신용 (열쇠파케트다음에 나타난다. 사용자가 정의)	공개열쇠/사용자ID쌍에 부여된 신용 (사용자ID파케트뒤에 있다. PGP에 의한 계산)	서명에 부여된 신용 (서명묶음다음에 놓인다.이서명자에 대한 OWNERTRUST의 캐시된 복사)
OWNERTRUST 마당 – 정의되지 않은 신용 – 알려 지지 않은 사용자 – 다음 열쇠들을 서명하는데 보통 신용되지 않는다. – 다른 열쇠들을 서명하는데 항상 신용된다. – 이 열쇠는 비밀열쇠고리에 존재한다. BUCKSTOP비트 – 이 열쇠가 비밀열쇠고리에 나타나면 설정된다.	KEYLEGIT마당 – 알려 지지 않거나 정의되지 않은 신용 – 신용되지 않은 열쇠소유권 – 열쇠소유권에 대한 별도의 신용 – 열쇠소유권에 대한 완전신용 WARNONLY 비트 – 완전유효성검증이 되지 않은 열쇄가 암호화에 리용될 때 사용자에게 경고만을 위해 설정된다.	SIGTRUST마당 – 정의되지 않은 신용 – 알려 지지 않은 사용자 – 흔히 다른 열쇠들로 서명하는데 신용되지 않는다. – 항상 다른 열쇠들로 서명하는데 신용된다. – 이 열쇠는 비밀열쇠고리에 있다(최종신용). CONTIG 비트 – 서명에 의해 련속적인 신용증명경로가 최종적으로 신용된 열쇠고리소유자에게 되돌이인도될 때 설정된다.

사용자 A의 공개열쇠고리를 취급한다고 가정하자. 이때 그 신용처리조작을 다음과 같이 서술할수 있다.

1. A가 세 공개열쇠를 공개열쇠고리에 삽입할 때 PGP는 이 공개열쇠들의 소유자와

관련한 신용기발에 어떤 값들을 할당하여야 한다. 소유자가 A이고 따라서 이 공개 열쇠가 비밀열쇠고리에도 있으면 최종신용의 값은 자동적으로 신용마당에 할당된다. 그렇지 않고 PGP가 A에게 이 열쇠의 소유자에게 할당될 신용에 대한 그의 평가를 요구하면 A는 요구하는 수준을 넣어야 한다. 사용자는 이 소유자를 알지 못한다,믿지 못한다, 겨우 믿는다 또는 완전히 믿는다는 등으로 특징 지울수 있다.

2. 새로운 공개열쇠가 들어 가면 하나 또는 그 이상의 서명들이 거기에 붙을수 있다. 후에 몇개의 서명들이 더 부과될수도 있다. 어떤 서명이 입력에 삽입될 때 PGP는 공개열쇠고리를 탐색하여 이 서명의 발행자가 알려 진 공개열쇠소유자들속에 있는가를 본다. 만일 그렇다면 그 소유자에 대한 OWNERTRUST값이 이 서명의 SIG신용마당에 부여 된다. 그렇지 않으면 unknownuser값이 부여된다.
3. 열쇠정당성마당의 값은 이 입력에 존재하는 서명신용마당에 기초하여 계산된다. 만일 적어도 하나의 서명이 최종의 서명신용값을 가지면 열쇠정당성마당값은 완전히 설정된다. 1/X의 무게는 항상 신용되는 서명들에 주어 지며 1/Y의 무게는 보통 신용되는 서명들에 주어 진다. 여기서 X와 Y는 사용자-구성가능파라메터들이다. 열쇠/사용자결합의 안내자들의 전체 무게가 1에 도달하면 속박은 믿을만하다고 인정되며 열쇠정당성값은 완전히 설정된다. 이리하여 최종신용이 없이 늘 신용되는 X개의 서명들이나 보통 신용되는 Y개의 서명들 또는 일부 결합이 필요하게 된다.

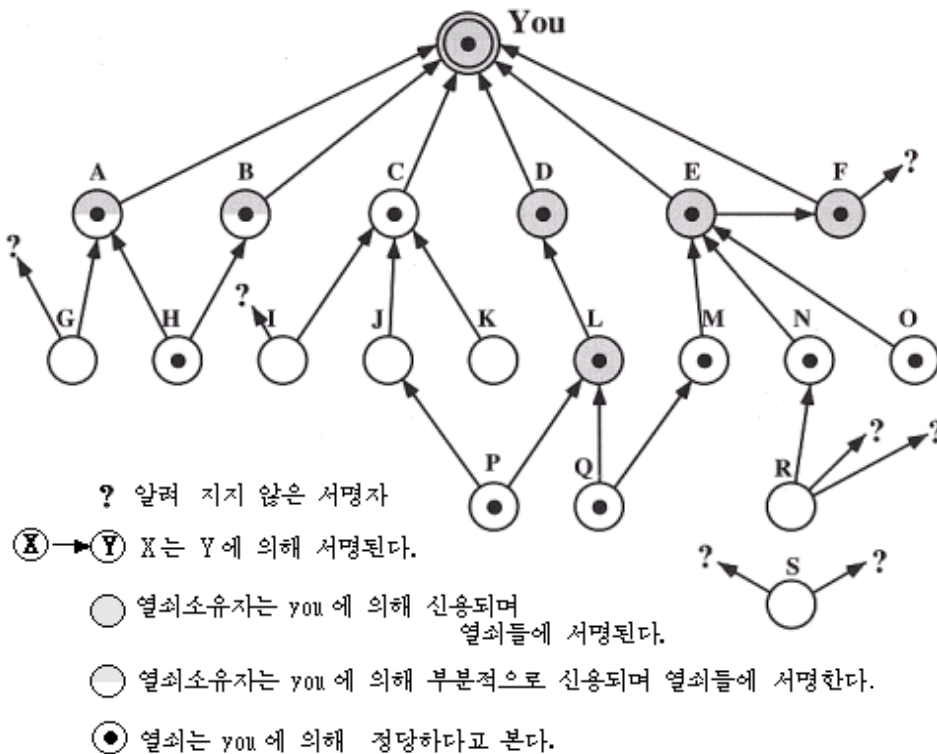



그림 12-7. PGP신용모형실례

주기적으로 PGP는 공개열쇠고리를 처리하여 일치성을 보장한다. 이것은 본질적으로 내리처리과정이다. 매개 OWNERTRUST마당들에 대하여 PGP는 그 소유자에 의해 보증된 모든 서명들에 대해 고리를 조사하고 OWNERTRUST마당과 같은 SIG신용마당을 갱신한다. 이 과정은 최종신용이 있는 열쇠들로부터 시작된다. 다음 모든 KEYLEGIT마당들은 부가된 서명들에 기초하여 계산된다.

그림 12-7에 서명신용과 열쇠정당성의 관련방식에 대한 실례를 주었다. 그림에는 공개열쇠고리의 구조를 보여 주었다. 사용자는 여러개의 공개열쇠들을 알고 있는데 일부는 그 소유자들로부터 직접 알며 일부는 열쇠봉사기와 같은 제3자로부터 안다.

"You"로 표시된 마디는 그 사용자에 대응하는 공개열쇠고리에로의 입력을 의미한다. 이 열쇠는 합법적이며 OWNERTRUST값은 최종신용이다. 열쇠고리의 서로 다른 매 마디들은 사용자에 의해 어떤 다른 값들이 할당되지 않는 한 정의되지 않은 OWNERTRUST을 가진다. 이 실례에서 사용자는 다른 열쇠들에 서명하기 위해 다음의 사용자들 즉 D, E, F를 항상 믿는다고 하였다. 사용자는 다른 열쇠들에 서명하기 위해 사용자 A와 B를 부분적으로 신용한다.

따라서 그림 12-7에서 마디들의 어둡고밝은 정도는 이 사용자에 의해서 부여되는 신용수준을 가리킨다. 그 나무구조는 어느 열쇠들이 어떤 다른 사용자들에 의해 서명되는가를 가리킨다. 만일 어떤 열쇠가 그것을 열쇠고리에 가지고 있는 사용자에 의해 서명된다면 화살표식은 서명된 열쇠를 그 서명자에게 련결한다. 만일 열쇠가 그것을 열쇠고리에 가지고 있지 않는 사용자에 의해 서명된다면 화살표식은 서명된 열쇠를 물음표에 련결한다.

몇가지 문제들을 그림 12-7에서 례증하였다.

1. 이 사용자에 의해 완전히 또는 부분적으로 신용되는 소유자들의 모든 열쇠들은 마디 L을 내놓고 그 사용자에 의해 서명된다는것을 주의해 둔다. 마디 L의 존재가 가리키는것처럼 이러한 사용자서명은 항상 필요한것은 아니지만 현실에서 대부분의 사용자들은 자기들이 신용하는 대부분의 소유자들의 열쇠들에 서명할수 있다. 실례로 E의 열쇠가 신용되는 안내자 F에 의해 이미 서명되였는데도 그 사용자는 바로 E의 열쇠를 선택하여 서명한다.
2. 어떤 열쇠를 확인하는데 두개의 부분적으로 신용되는 서명들이면 충분하다고 가정한다. 여기로부터 사용자 H의 열쇠는 부분적으로 신용되는 A와 B에 의해 서명되였으므로 PGP에 의해 합법적이라고 간주한다.
3. 열쇠는 완전히 신용되는 한명 또는 두명의 부분적으로 신용되는 서명자들에 의해 서명되면 합법적이라고 결정되지만 그 사용자는 다른 열쇠들에 서명하는데 신용되지 않을수도 있다. 실례로 N의 열쇠는 사용자가 신용하는 E에 의해 서명되므로 정당하지만 N에 그 신용값을 할당하지 않았으므로 다른 열쇠들에 서명하는데 신용되지 않는다. 따라서 R의 열쇠가 N에 의해 서명된다고 해도 PGP는 R의 열쇠가 합법적이라고 간주하지 않는다. 만일 개인통보문을 어떤 개별적사람에게 보내려고 한다면 모든 측면에서 그 사람을 믿지 않아도 된다. 다만 그 사람에 대해 정확한 공개열쇠을 가지고 있다는것만을 확신하는것이 필요하다.
4. 그림 12-7에 두개의 알려 지지 않은 서명을 가지는 분리된 《고아》마디 S의 실례를 보여 주었다. 이러한 열쇠는 열쇠봉사기로부터 얻을수 있다. PGP는 이 열쇠가 공개된 봉사기로부터 입수한것이므로 쉽게 합법적이라고 가정할수 없다. 그 사용자는 열쇠에 서명하거나 PGP에 그 열쇠의 서명자들중의 하나를 완전히 신용한다는것을 알리여 그 열쇠의 합법칙성을 선언하여야 한다.

마지막문제: 앞에서 다중사용자ID들이 공개열쇠고리중의 하나의 공개열쇠와 관련될 수 있다는것이 언급되였다. 이것은 어떤 사람이 이름을 변화시키거나 한 사람이 다른 전자우편주소들을 가리키는 다중이름으로 서명을 하여 안내될수 있기때문이다.공개열쇠를 나무의 뿌리로 생각할수 있다. 공개열쇠에는 그와 관련한 여러개의 사용자ID_S가 있는데 매 사용자ID밑에 또 여러개의 서명들이 있다. 열쇠에 대한 개별적사용자ID의 속박은 그 사용자ID와 열쇠와 관련한 서명에 의존한다.

공개열쇠취소

사용자는 자기의 현재공개열쇠를 취소할수 있다.왜냐하면 가로채기가 걱정되거나 단순히 한 열쇠의 사용기간이 확장되는것을 피하려 할수 있기때문이다. 적이 암호화되지 않은 비밀열쇠의 복사를 얻거나 비밀열쇠고리로부터 비밀열쇠와 통과단계(passphrase)를 다 얻을수 있는 경우를 가상한다.

공개열쇠의 취소는 소유자가 자기가 서명한 열쇠취소증명서를 발급하는것으로 규정된다. 이 증명서는 일반서명증명서와 같은데 다른것은 다만 증명서의 목적이 이 공개열쇠의 리용을 페지한다는 지적자를 포함한다는것이다.해당한 비밀열쇠를 사용하여 공개열쇠를 취소하는 증명서에 서명할수 있다.소유자는 공개열쇠고리를 갱신할수 있도록 가능한껏 빨리 그리고 널리 이 증명서를 류포시켜야 한다.소유자는 그때 지난 기간의 거래자들도 자기들의 공개열쇠고리들을 갱신할수 있도록 될수록 널리 그리고 빨리 이 증명서를 류포시키려고 할것이다.

어떤 소유자의 비밀열쇠를 절취한 적도 이러한 증명서를 발급할수 있다는데 주의한다. 그러나 이것은 적이나 합법적소유자가 공개열쇠를 사용하는것을 부정하므로 도적질당한 비밀열쇠의 악의 있는 사용보다는 훨씬 덜 위협적일것이다.

12.2 S/MIME

S/MIME(Secure/Multipurpose Internet Mail Extension)은 RSA자료보안기술에 기초한 MIME인터네트전자우편형식의 규격으로 보안을 강화한것이다. PGP와 S/MIME가 둘다 IETF규격화되여 있지만 S/MIME는 상업 및 기관적인 용도의 공업규격으로서 등장하는 반면에 PGP는 많은 사용자들에 대하여 개인전자우편보안수단으로만 유지될것이다.

S/MIME를 리해하기 위하여 먼저 그것이 기초한 전자우편형식(즉 MIME)에 대한 일반적리해를 가지는것이 필요하다. 그러나 MIME의 의미를 리해하자면 지금도 보통 쓰고 있는 전통적인 전자우편형식규격 RFC를 리해하는것이 필요하다. 따라서 이 절에서는 먼저 이 두개의 초기규격들에 대한 소개를 주고 S/MIME에 대하여 론의하기로 한다.

RFC 822

RFC 822는 전자우편을 통해 보내오는 본문통보문의 형식을 정의한다. 그것은 인터네트에 기초한 본문우편통보문에 대한 규격이며 현재도 널리 쓰이고 있다. RFC 822에서는 통보문들을 봉투와 내용부들을 가지고 있는것으로 본다. 봉투에 전송과 배달을 실현하는데 필요한 어떤 정보가 들어 있다. 내용부는 수신측에 배달되여야 할 객체로 구성된다. RFC 822규격은 내용부에 한해서만 적용된다. 그러나 내용규격에는 우편체계에 의하여 봉투를 만드는데 리용될수 있는 머리부마당들이 포함되며 규격화는 프로그람적으

로 이러한 정보의 획득을 도모하는것을 목적으로 한다.

RFC 822에 따르는 통보문의 총적구조는 매우 간단하다. 통보문은 몇개의 머리부행(머리부)과 그 다음의 제한 없는 본문(본체)으로 구성된다. 머리부는 빈 행으로 본체와 분리된다. 다시말하여 통보문은 ASCⅡ본문이며 첫 빈 행우의 모든 행들은 우편체계의 사용자대표부로 쓰이는 머리부행들이라고 가정된다.

머리부행은 보통 예약어다음에 두점, 그 다음의 예약어변수들로 작성된다. 그 형식은 긴 행을 몇개의 행들로 가를수 있게 되여 있다. 제일 많이 리용하는 예약어들로는 From, To, Subject 및 Data이다.

여기에 실례통보문이 있다.

Data: Tue, 16 Jan 1998 10: 37:17(EST)
From: “William Stallings” <ws@shore.net>
Subject: The Syntax in RFC 822
To: Smith @Other-host.com
Cc: Jones @Yet-Another-Host.com
Hello. This section begins the actual message body, which is delimited form the message heading by a blank line.

RFC 822머리부들에서 공통적으로 흔히 볼수 있는 또다른 마당은 Message-ID이다. 이 마당은 통보문과 관련한 유일한 식별자를 포함한다.

다목적인터네트우편확장(Multipurpose Internet Mail Extensions)

MIME는 단순우편전송계약(Simple Mail Transter Protocol :SMTP) 또는 다른 우편전송규약 및 전자우편용의 RFC 822리용의 제반 문제들과 제한성들을 취급하기 위한 RFC 822의 확장이다.

1. SMTP는 실행형파일들이나 다른 2진객체들을 전송할수 없다. 많은 방식들은 2진파일들을 대중적인 UNIXUU부호화/UU복호방식을 비롯하여 SMTP우편체계들이 리용될수 있는 본문형태로 변환하는데 리용하고 있다. 그러나 이것들은 하나도 규격화되여 있지 않다.
2. SMTP는 민족어문자들이 들어 있는 본문자료를 전송할수 없다. 왜냐하면 이것들은 십진128의 값을 8-bit부호나 그이상의 부호로 표현하며 SMTP는 7-bitASCⅡ로 제한되기때문이다.
3. SMTP봉사기들은 어떤 크기이상의 우편통보문들은 거부한다.
4. ASCⅡ와 문자부호 EBCDIC사이를 번역하는 SMTP관문들은 일관성 있는 온전한 넘기기들을 리용하지 않는것으로 해서 번역에서 문제들을 발생한다.
5. X.400전자우편망에 대한 SMTP관문들은 X.400통보문들에 포함되는 비본문화자료들을 조종할수 없다.
6. 일부 SMTP실현들에서는 RFC 821에서 정의되는 SMTP규격들을 완전히 준수하지 않는다. 공통적인 문제들로는 다음의것들이 있다.
   - 삭제, 부가 또는 행바꾸기와 행간조절의 재배렬
   - 76문자이상의 행들을 자르거나 겹치기
   - 공백(타브 및 공백문자)을 제거

- 통보문의 행들을 같은 길이로 메꾸기
- 타브문자들을 여러가지 공백문자들로 변환

MIME의 목적은 이 문제들을 현재의 RFC 822의 실현들과 호환하는 방법으로 해결하는것이다.

그 명세서가 RFC_S2045-2048에 제시되였다.

개요

MIME명세서에는 다음의 요소들이 포함된다.

1. 다섯개의 새로운 통보문머리부마당들이 정의되는데 그것들은 RFC 822머리부에 포함될수 있다. 이 마당들은 그 통보문의 본체에 대한 정보를 제공한다.
2. 많은 내용부형식들이 정의되며 따라서 다매체전자우편을 지원하는 표현들이 규격화된다.
3. 임의의 내용부형식을 그 우편체계에 의한 교체로부터 보호되는 형식으로 변환하는 전송부호화가 정의된다.

이 소절에서는 다섯개의 통보문머리부마당들을 소개한다. 다음의 두 소절들에서는 내용부형식들과 전송부호화를 취급한다.

다섯개의 머리부마당들은 MIME에서 다음과 같이 정의된다.

- **MIME판본**(MIME-Version): 대부분이 파라메터값 1.0을 가진다. 이 마당은 그 통보문이 RFC_S2045-2046에 적합하다는것을 가리킨다.
- **내용부형식**(Content-Type): 수신자가 적당한 중개자나 꾸밈새를 선택하여 그 사용자에게 자료를 제출하거나 그렇지 않으면 적당한 방법으로 그 자료를 취급하는 통보문본체에 들어 있는 자료를 충분히 자세히 서술한다.
- **내용부전송부호화**(Content-Transfer-Encoding):우편전송에 적당한 방법으로 통보문의 본체를 표현하는데 쓰이는 전송형태를 지적한다.
- **내용부ID**(Content ID):여러 정황들에서 MIME실체들을 유일하게 확인하는데 쓰인다.
- **내용부서술**(Content-Description):본체를 가지는 객체의 본문서술. 이것은 객체가 읽을수 없을 때(즉 음성자료) 유용하다.

이 마당들의 일부 또는 모두가 규격화된 RFC 822머리부에 나타날수 있다. 편리한 실장이 되자면 MIME-Version, Content-Type 및 Content-Transfer-Encoding마당들을 지원해야 한다. 즉 Content-ID와 Content-Description마당들은 선택적이며 수신실현에 의해 무시될수 있다.

MIME 내용부형

대부분의 MIME명세는 여러가지 형태의 내용부형태들에 대한 정의와 관련된다. 여기에는 다매체환경에서 여러가지 정보표현을 취급하는 규격화된 방법들을 제공할데 대한 요구가 반영되여 있다.

표 12-3에 RFC 2046에 명기된 내용부형들을 보여 주었다. 여기에 7개의 서로 다른 기본형태의 내용부들과 모두 15개의 부분형태들이 있다. 일반적으로 내용부형은 일반형태의 자료를 선언하고 보조부분형(subtype)은 그 자료형태에 대한 특정의 형식을 명기한다.

본체의 본문형에 대한 지적된 문자모임의 지원을 제외하고는 본문의 내용을 충분히 리해하는데 특별한 쏘프트웨어가 요구되지 않는다. 주보조부분형은 평문인데 그것은 단순히 ASCII문자들의 렬 또는 ISO 8859문자들의 렬이다. 보강된 부분형은 더 큰 형식화 유연성을 준다.

다중부분형(Multipart type)은 본체가 다중독립인 부분들을 포함한다는것을 나타낸다. Content-Type머리부마당은 이른바 경계파라메터를 포함하는데 그것은 본체부분들사이의 경계를 정의한다. 매 경계들은 새로운 행으로 시작되며 두개의 련결부호와 그 다음의 경계값으로 구성된다. 마지막부분의 끝을 가리키는 마지막경계도 역시 두개의 련결부호 《-》의 뒤붙이를 가진다. 매개 부분에는 선택적인 보통의 MUME머리부가 있을수 있다.

간단한 본문으로 구성된 두개의 부분들을 포함하는 다중부분통보문에 대한 간단한 실례가 있다(RFC 2046).

표 12-3. MIME내용부형

형	부분형	서술
본문	평문 보강	비형식화된 본문: ASCII나 ISO 8859이다. 더 큰 형식화유연성을 제공한다.
다중부분	혼합형	서로 다른 부분들은 독립이나 함께 전송된다. 그것들은 우편통보문에 나타나기 위해 수신자에게 제출되여야 한다.
	병렬	부분들을 수신자에게 배달하기 위한 순서가 정의되지 않는데서만 Mixed와 다르다.
	대안	서로 다른 part들은 같은 정보에 대한 대안적인 판본들이다. 그것들은 원본에 대한 충실성이 증가하도록 배렬되며 수신측의 우편체계는 《제일 좋은》판본을 사용자에게 현시한다.
	요약정보	Mixed와 비슷하지만 매 부분의 기존형/부분형은 통보문/rfc822이다.
통보문	rfc822 partial 외부-본체	본체 그자체는 RFC 822에 적합한 교갑화된 통보문이다. 큰 우편항목들을 수신측에 투명하도록 토막화하는데 리용 다른곳에 존재하는 대상에 대한 지시기를 포함한다.
화상	jpeg Gif	이 화상은 JPEG형식으로 JFIF부호화된다. 화상은 GIF형식이다.
비데오	mpeg	MPEG형식
음성	Basic	8kHz의 본보기비률로 한 통로8-bit ISDN mu-law부호화
응용	PostScript octet-stream	Adobe PostScript 8-bit바이트들로 구성된 일반 2진자료

Form: Nathaniel Boreustein <nsb@bellcore.com>
To: Ned Freed<ned@innosoft.com>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed;boundary= “simple boundary”

This is implicitly typed plain ASCII text. It does NOT end with a linebreak
Composers to include an explanatory note to non-MIME conformant readers.
--simple boundary
This is the preamble. It is to be ignored, though it is a handy place for mail
--simple boundary
Content-type: text/plain: charset=us-ascii

This is explicitly typed plain ASCII text. It DOES end with a linebreak

--simple boundary
This is the epilogue. It is also to be ignored

다중부분형에 4개의 보조부분형들이 있는데 그 모두는 총적으로 같은 문법을 가진다. Multipart/mixed subtype는 특정의 규칙으로 묶을 필요가 있는 다중독립본체부분들이 있을 때 리용된다. Multipart/parallel subtype에서 부분들의 순서는 상관 없다. 수신체계가 정당하면 다중부분들을 병렬로 표시할수 있다. 실례로 화상이나 본문부분에는 화상이나 본문이 현시되는 동안 음성주석이 동반될수 있다.

multipart/alternative subtype에서 여러 부분들은 같은 정보에 대한 서로 다른 표현들이다. 그에 대한 다음과 같은 실례가 있다.

From: Nathaniel Borenstein <nsb@ bellcore.com>
To: Ned Freed <ned@innosoft. com>
Subject: Formatted text mail
MIME-Version : 1.0
Content-Type: multipart/alternative; boundary=boundary 42

- -boundary

Content-Type: text/plain;charset=us-ascii

...plain text version of message goes here...

- -boundary 42
Content-Type: text/enriched

...RFC 1896 text/enriched version of same message goes here...
- -boundary42 - -

이 보조부분형에서 본체부분들은 우선권순서로 배렬된다. 실례에서 만일 수신체계가 text/enriched형식으로 통보문을 현시할수 있으면 그렇게 되고 그렇지 않으면 평문형식이 리용된다.

다중부분/요약정보부분형은 매 본체부분들이 머리부들을 가진 RFC 822통보문이라

고 판별될 때 리용된다. 이 보조부분형을 리용하면 부분들이 개별적통보문들인 통보문의 구조를 쉽게 구성할수 있다. 실례로 그룹의 중재자는 관계자들로부터 전자우편통보문들을 수집하고 이 통보문들을 묶은 다음 그것들을 한개의 교갑화된 MIME통보문으로 하여 보낼수 있다.

통보문형은 MIME의 많은 중요 기능들을 제공한다. message/rfc822 subtype는 머리부와 본체를 포함하는 전체 통보문이 본체라는것을 지적한다. 이 보조부분형의 이름과 교갑화된 통보문은 단순한 RFC 822통보문만이 아니라 임의의 MIME통보문일수도 있다.

통보문/보조부분형을 쓰면 큰 통보문을 여러개의 부분들로 쉽게 구분할수 있는데 그것은 목적지에서 재결합되여야 한다. 이 부분형에 대하여 세개의 파라메터들이 내용형에 렬거된다. 즉 통보문/부분마당, 같은 통보문의 모든 토막들에 대한 id공유권, 매 토막들에 유일한 렬번호 및 전체 토막수

통보문/외부본체부분형은 이 통보문으로써 운반되는 실지자료가 본체에 포함되지 않는다는것을 가리킨다. 대신 본체에는 그 자료에 접근하는데 필요한 정보가 들어 있다. 다른 통보문형들과 마찬가지로 message/external-body부분형은 바깥머리부와 그자체의 머리부를 가지는 교갑화된 통보문을 가진다. 바깥머리부에 필요한 마당은 바로 내용부마당형마당인데 이것을 message/external-body보조부분형으로 정의한다. 내부머리부는 교갑화통보문에 대한 통보문머리부이다. 외부머리부의 Content-Type마당에는 접근형식파라메터가 포함되게 되는데 이것은 FTP(file transfer protocol)와 같은 접근방법을 가리킨다.

응용형은 다른 종류의 자료, 대표적으로 해석되지 않은 2진자료나 우편응용프로그람에 의해 처리되는 정보를 가리킨다.

MIME Transfer Encoding

내용부형명세서외에 MIME명세서의 다른 중요한 요소는 통보문본체들에 대한 전송부호화의 정의이다. 목적은 최대한 넓은 범위의 환경들에서 믿음직한 배달을 제공하는것이다.

MIME규격은 자료를 부호화하는 두가지 방법을 정의한다. 내용부전송부호화마당은 실지로 표 12-4에서와 같이 6개의 값을 가질수 있다. 그러나 이 값들중 세개(7bit, 8bit 및 2bit)는 부호화가 진행되지 않았다는것을 가리키며 본래자료에 대한 일부 정보를 제공한다. SMTP전송에서 7-bit형식을 리용하는것이 간편하다. 8-bit와 2-bit형식들은 다른 우편전송과 관련하여 리용할수 있다. 다른 내용부전송부호화값은 x-통표인데 이것은

표 12-4. MIME Transfer Encodings

7bit	자료는 모두 ASCⅡ문자들의 짧은 행들에 의해 표현된다.
8bit	행들은 짧지만 비ASCⅡ문자들이 있을수 있다.
2진수	많은 비ASCⅡ문자들이 존재할뿐아니라 행들이 SMTP전송에 맞게 짧지는 않다.
Quoted-printable	부호화되는 자료가 대부분 ASCII본문이면 자료의 부호화된 형식이 여전히 사람이 충분히 인식할수 있도록 자료를 부호화한다.
Base64	입력의 6-bit블로크들을 출력의 8-bit블로크들에로 넘기기하여 자료를 부호화하는데 그것들은 인쇄가능한 ASCII문자들이다.
x-token	명명된 비규격부호화

어떤 다른 부호화방식이 리용된다는것을 가리키며 이름이 제공된다. 그것이 vendor-specific 또는 application-specific방식일수 있다. 정의된 두 실지부호화방식들은 quoted-printable과 base64이다. 두개의 방식들은 본질적으로 사람이 읽기 쉬운 전송기술과 타당하게 조밀화하여 자료의 모든 형들에 대하여 안전한 전송기술사이의 선택을 제공하기 위해 정의된다.

quoted-printable전송부호화는 그 자료가 인쇄가능한 ASCII문자들에 부합되는 옥테드들로 구성될 때 유용하다. 본질적으로 그것은 문자들의 부호의 16진표현으로 불안전문자들을 표시하며 76문자로 통보문행들을 제한하는 취소가능(연한)행 구분들을 도입한다.

Radix-64부호화로 알려 진 **base64전송부호화**는 임의의 2진자료를 우편전송프로그람에 의한 처리에 모순되지 않는 타당한 방법으로 부호화하는 일반방법이다. 그것은 PGP에서 리용되는데 부록 12-2에 소개한다.

다중부분실례

FRC 1521에서의 그림 12-8에서는 복합다중부분통보문에 대한 륜곽을 보여 주었다. 통보문은 직렬로 현시되는 다섯개의 부분들 즉 2개의 안내평문부분들, 매몰된 다중부분통보문, 덧본문부분 및 비ASCII문자들로 결속되는 교갑화본문통보문으로 되여 있다. 매몰된 다중부분통보문은 병렬로 현시되는 두개의 부분들 즉 화상과 음성쪼각을 가진다.

규범적인 형식(Canonical Form)

MIME와 S/MIME에서 중요한 개념은 규범적형식이다. 규범적형식은 내용부형에 적합한 체계들사이에서 리용하기 위하여 규격화된 형식이다. 이것은 본래형과 반대인데 특정의 체계에 고유한 형식이다. RFC 2049의 표 12-5를 통해 이 문제를 명백히 할수 있다.

S/MIME기능

일반기능적으로 S/MIME는 PGP와 매우 류사하다. 량자가 배타적론리합암호통보문들에 서명하는 기능을 제공한다. 이 소절에서는 S/MIME기능을 간단히 개괄한다. 그 다음 통보문형식들과 통보문준비를 통하여 이 기능을 더 자세히 조사한다.

기능

S/MIME는 다음의 기능들을 제공한다.

- **봉투화된 자료**: 이것은 임의의 형의 암호화된 내용과 하나 또는 그이상의 수신측들에 대한 암호화된 암호열쇠들로 이루어 진다.
- **서명된 자료**: 수자서명은 서명되여야 할 내용부의 통보문요약정보를 취하고 그것을 서명자의 비밀열쇠로 암호화하여 얻어 진다. 서명은 내용부와 함께 base64부호로 부호화된다. 서명된 자료통보문은 수신측만이 S/MIME기능으로 볼수 있다.
- **열린 서명자료**: 서명된 자료와 마찬가지로 내용의 수자서명이 형성된다. 그러나 이 경우에 수자서명만이 base64로 부호화된다. 결과로서 수신측들은 S/MIME기능이 없이 그 서명을 검증할수 없다고 해도 통보문내용은 볼수 있다.
- **서명봉투화된 자료**: 서명전용실체와 암호전용실체들은 서로 포개 져 암호화된 자료가 서명되거나 서명된 자료 혹은 명백한 서명자료가 암호화될수 있다.

From: Nathaniel Borenstein <nsb@ bellcore.com>
To: Ned Freed <ned@innosoft. icom>
Subject: A multipart example
Content-Type: multipart/mixed:
  boundary=unique-boundary-1

이것은 다중부분통보문의 머리말부분이다. 다중부분형식을 알고 있는 우편열람자들은 이 머리말을 무시할수 있다. 이 본문을 읽으면서 독자들은 다중부분통보문들을 적당히 현시하는 방법을 리해하는 우편열람자로 되는 과정을 고찰하고 싶을것이다.

Unique-boundary-1

...Some text appears here...
[Note that the preceding blank line means no header fields were given and this is text, with charset US ASCII. It could have been done with explicit typing as in the next part.]

--unique-boundary-1
Content-type: text/plain;charset=US-ASCII

This could have been part of the previous part, but illustrates explicit versus implicit typing of body parts.

--unique-boundary-2
Content-Type:audio/basic
Content-Transfer-Encoding:base64

...base64-encoded 8000Hz single-channel mu-law-format audio data goes here...

--unique-boundary-2
Content-Type:image/jpeg
Content-Transfer-Encoding:base64

...base64-encoded image data goes here...

--unique-boundary2--

--unique-boundary-1
Content-type:text/enriched

This is<bold><italic>richtext.</italic></bold><smaller>as defined in RFC 1896</smaller>
Isn't it<bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822

Form:mailbox in US-ASCII)
To:(address in US-ASCII)
Subject:(subject in US - ASCII)
Content-Type:Text/plain;charset=ISO-8859-1
Content-Transfer-Encoding:Quoted-printable

...Additional text in ISO-8859-1 goes here...

--unique-boundary-1--

그림 12-8. MIME통보문구조실례

표 12-5. 본래의 형식과 규범적형식(Native and Canonical Form)

문자의 형식	전송할 본체는 체계의 본래의 형식으로 창조된다. 본래의 문자모임이 리용되는데 적당한 곳에서는 국부적행마감규정들도 리용된다. 본체는 UNIX-방식본문파일이거나 Sun raster화상, UMS첨수화된 파일, 기억기에만 기억된 체계의존형식의 음성자료 그밖에 정보의 어떤 형식의 포함에 대한 국부모형일수 있다. 기본적으로 자료는 매체형에 의해 명기된 형에 일치하는 《본래(native)》형식으로 창조된다.
규범적형식	레부호길이와 같은 《대역밖의》정보나 파일특성정보를 포함하는 전체 본체는 보편적인 규범적형식으로 변환된다. 본체의 특수한 매체형이나 그것의 관련된 속성들은 리용되는 규범적형식의 특성을 나타낸다. 문자모임변환, 음성자료의 변환, 압축 또는 여러가지 매체형들에 대한 여러가지 다른 특수한 조작들에 의해 적합한 규범적형태로 변환된다. 만일 문자모임변환이 포함되면 그 매체형에 대한 의미를 리해하는데 주의해야 하는데 그것은 임의의 문자모임변환에 대한 강한 함축을 가질수 있다.

암호알고리듬

표 12-6에 S/MIME에서 리용하는 암호알고리듬들을 개괄하였다. S/MIME는 RFC 2199로부터 취한 다음의 용어들을 리용한다.

- MUST:명세서의 절대적요구로 정의된다. 실현은 이 특성이나 그 명세서와 일치하는 기능을 포함해야 한다.
- SHOULD: 특정한 환경에서 이 특성이나 기능을 무시할수 있는 정당한 리유가 있을수 있으나 실현에 특성이나 기능이 포함되는것이 좋다.

표 12-6. S/MIME에서 리용되는 암호화알고리듬

기능	요구조건
수자서명을 구성하는데 리용되는 통보문요약정보를 창조한다.	SHA-1과 MD5를 지원해야 한다(MUST). SHA-1을 리용한다(SHOULD).
통보문요약정보를 암호화하여 수자서명을 구성한다.	송신 및 수신중개자들은 DSS를 지원하여야 한다(MUST). 송신중개자는 RSA암호화를 지원한다. 수신중개자들은 열쇠크기 512-bit～1024-bit열쇠크기를 가지는 RSA서명의 검증을 지원한다(SHOULD).
통보문전송을 위한 대화열쇠를 암호화한다.	송신 및 수신중개자들은 디피-헬만(Diffie-Hellman)을 지원하여야 한다(MUST). 송신중개자는 열쇠크기가 512-1024bit인 KSA암호화를 지원한다(SHOULD). 수신중개자는 RSA복호를 지원한다(SHOULD).
전송을 위하여 통보문을 한번만 쓰는 대화열쇠로 암호화한다.	송신중개자들은 3중DES나 RC2/40로 암호화를 지원한다(SHOULD). 수신중개자들은 3중DES를 리용하여 복호를 지원하여야 하며 RC2/40으로 복호를 지원한다(MUST).

S/MIME는 세개의 공개열쇠알고리듬을 통합한다. 수자서명규격(DSS)은 보다는 수자서명을 위하여 리용되는 알고리듬이다. S/MIME는 대화열쇠를 암호화하기 위하여 등용된 알고리듬으로서 디피-헬만(Diffie-Hellman)을 제공하는데 사실 ElGamal(문제 6-19를 볼것)은 알려 진 암호화/복호를 제공하는 디피-헬만(Diffie-Hellman)의 변종을 리용한다.

대안으로서 5장에서 서술된 RSA는 서명과 대화열쇠암호화에 리용할수 있다. 이것들은 PGP에서 리용되는 알고리듬과 같은것들이며 높은 수준의 보안을 제공한다. 수자서명을 창조하는데 쓰이는 하쉬함수에서 명세로는 160-bitSHA-1가 권고되나 128-bitMD5에 대한 지원을 요구한다. 3장에서 본것처럼 MD5의 보안에 대한 정당한 관계가 있으므로 명백히 SHA-1은 대안식으로 된다. 그러나 MD5가 광범히 실현되고 또 지원된다.

통보문암호화에 대해 3중DES가 권고되지만 편리한 실장들은 40-bitRC2을 지원한다. 후자는 약한 암호화알고리듬이지만 미국수출조종에는 부합된다.

S/MIME명세서에서는 어느 내용부암호화알고리듬을 리용하겠는가를 결정하는 수속에 대하여 론의한다. 본질적으로 송신중개자는 두가지의 결정을 한다. 첫째로, 송신중개자는 수신중개자가 주어 진 암호알고리듬으로 복호할수 있는가를 결정해야 한다. 둘째로, 만일 그 수신중개자만이 약하게 암호화된 내용을 접수할수 있다면 송신중개자는 약한 암호화로 보내는것이 접수가능한가를 결정하여야 한다. 이 결정과정을 지원하기 위해 송신중개자는 자기의 복호능력을 알릴수 있다. 수신중개자는 앞으로의 리용을 위해 그 정보를 보관할수 있다.

다음의 규칙들은 송신자에 의해 차례로 집행된다.

1. 만일 송신자가 의도된 수신으로부터 우선권복호능력들의 목록을 가지면 그 SHOULD는 그것이 리용할수 있는 목록상에서 첫번째(최우선권)능력을 선택한다.
2. 송신자가 의도한 수신으로부터 이런 기능들의 목록을 받지 못했지만 수신자로부터 하나 또는 그 이상의 통보문을 받았으면 나가는 SHOULD는 그 의도된 수신에서 접수된 최종서명이 있고 암호화된 통보문에서 리용되였던것과 같은 암호화알고리듬을 리용한다.
3. 만일 송신자가 의도하는 수신자의 복호기능들에 대한 지식을 가지지 못하고 수신자가 그 통보문을 복호하지 못할것 같으면 송신자(SHOULD)는 3중DES를 리용한다.
4. 만일 송신자가 의도하는 수신자의 복호기능들에 대한 지식을 가지지 못하고 그 통보문을 복호하지 못할것 같으면 송신자는 KC 240을 리용한다.

만일 어떤 통보문이 다중수신자들에게 보내지고 공통암호알고리듬이 그들에게 선택되지 않았다면 송신자는 두개의 통보문을 보낼 필요가 있다. 그러나 그 경우 보안이 부족한 복사물의 전송에 의하여 통보문의 보안이 약해 진다는데 주의하는것이 중요하다.

S/MIME통보문

S/MIME는 표 12-7에서 보여 준 많은 새로운 MIME내용부형들을 리용한다. 새로운 응용형들은 모두 PKCS를 리용한다. 이것은 RSA서고들에 의해 발행된 공개열쇠암호명

세서들의 모임을 의미한다.

그 매개를 S/MIME통보문작성을 위한 일반수속을 본 다음 차례로 설명한다.

표 12-7. S/MIME내용부형

형	부분형	smime파라메터	서술
다중부분	서명된		두개 부분들로 된 clear-signed 통보문: 한쪽은 통보문이고 다른쪽은 서명이다.
응용	pkcs7-mime	서명된 자료	서명된 S/MIME실체
	pkcs7-mime	봉투화된 자료	암호화된 S/MIME실체
	pkcs7-mime	서명된 자료를 생성하지 않는다.	공개열쇠증명서만을 포함하는 실체
	pkcs7-서명	—	다중부분/서명된 통보문의 서명부분의 내용부형
	pkcs7-mime	—	증명서등록요구통보문

MIME실체의 보호

S/MIME는 서명, 암호, 또는 둘다를 가진 MIME실체를 보장한다. MIME실체로는 전체 통보문(RFC 822머리부를 내놓고)이 될수 있거나 만일 MIME내용부형이 다중부분이면 MIME실체는 통보문의 하나 또는 그이상의 부분들이다. MIME실체는 MIME통보문작성을 위한 일반규칙에 따라 준비된다. 다음 알고리듬식별자나 증명서와 같은 일부 보안련관자료와 함께 MIME실체는 S/MIME에 의하여 처리되여 이른바 PKCS객체를 생성한다. PKCS객체는 통보문내용으로서 취급되며 MIME로 포장된다(적당한 MIME머리부들을 제공한다). 이 과정은 특정한 객체들을 고찰하므로 명백해야 하며 실례로 될수 있다.

매 경우에 보내오는 통보문은 규범적형태로 변환된다. 특히 주어 진 형과 보조부분형에 대하여 적당한 규범적형식이 통보문내용부에 리용된다.

다중부분통보문에 대하여 적당한 형태는 매 부분형에 대하여 리용된다.

전송부호화를 리용하자면 특별한 주의를 돌려야 한다. 대부분의 경우 보안알고리듬을 적용한 결과는 부분적으로나 전체적으로 임의의 2진자료로 표현되는 객체를 생성한다. 다음에 이것을 바깥MIME통보문으로 싸고 base64로 전송부호화를 진행한다.

그러나 다중부분의 서명된 통보문인 경우 보조부분들중 하나에서 통보문내용은 보안과정에 변화되지 않는다. 그 내용부가 7bit가 아닌 한 서명된 내용부가 교체될 위험이 없도록 base64나 quoted printable로 전송부호화되여야 한다.

이제 매개 S/MIME내용부형들을 보자.

봉투화된 자료

응용/pkcs7-mime부분형은 S/MIME처리의 4개 부류들중의 하나에 리용되는데 매개는 유일한 smime-형파라메터를 가진다. 매 경우에 객체라고 하는 결과의 실체는 기초부호화규칙(Basic Encoding Rules:BER)으로 알려 진 형식으로 표현되는데 그것은 ITU-T 권고 X.209에서 정의된다. BER형식은 임의의 옥테드기호렬들로 구성되며 따라서 2진자료이다. 이러한 객체는 외부MIME통보문에서 base64로 전송부호화되여야 한다. 먼저 봉

투화된 자료를 보자.

봉투화된 자료 MIME실체를 준비하는 단계들은 다음과 같다.

1. 개개의 대칭암호알고리듬(RC 2140이나 3중DES)에 대하여 준우연대화열쇠를 생성한다.
2. 매 수신에 대하여 그 수신자의 공개RSA열쇠로 대화열쇠를 암호화한다.
3. 매 수신자에 대하여 송신자의 공개열쇠증명서, 대화열쇠를 암호화하는데 사용되는 알고리듬의 식별자 및 암호화된 대화열쇠를 포함하는 수신자정보로 알려 진 블로크를 준비한다.
4. 통보문내용을 대화열쇠로 암호화한다.

암호화된 내용부앞의 수신자정보(Recipient Info)블로크들은 봉투화된 자료(enveloped Data)를 구성한다. 다음 이 정보는 base64로 부호화된다. 본보기통보문(RFC 822머리부들을 제외)은 다음과 같다.

```
Content-Type: application/pkcs7-mime: smime-type=enveloped-data;
Name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment;filename=smime.p7m
```


rfvbnj756tbBhyHhHUujhJhjH77n8HHGT9HG4VQPfyF467GIGfHfYT6
7n8HHGghyHhHUujhJh4VqpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
OghIGfHfQbnj756YT64V

암호화된 통보문을 회복하기 위해 수신자는 먼저 base64부호화를 해제해야 한다. 다음 자기의 비밀열쇠를 써서 대화열쇠를 회복한다. 마지막으로 통보문내용을 그 대화열쇠로 복호한다.

서명자료(Signed Data)

Signed Data smime-type는 하나 또는 그이상의 서명자들이 리용할수 있다. 명확히 하기 위해 설명을 단일수자서명의 경우에 국한시켜 보자. EnvelopedData MIME실체를 준비하는 단계들은 다음과 같다.

1. 통보문요약(요약정보)알고리듬(SHA 또는 MD5)을 선택한다.
2. 서명되는 내용부의 통보문요약정보나 하쉬함수를 계산한다.
3. 통보문요약정보를 서명자의 비밀열쇠로 암호화한다.
4. 서명자의 공개열쇠증명서, 통보문요약알고리듬의 식별자,통보문요약정보를 암호화하는데 리용되는 알고리듬의 식별자 및 암호화된 통보문의 요약을 포함하는 서명자정보로 알려 진 블로크를 준비한다.

서명자정보실체는 통보문요약알고리듬식별자, 서명된 통보문 및 서명자정보를 포함

하는 블로크들의 렬들로 구성된다. 서명자료실체는 인정된 뿌리 또는 정점준위의 증명국으로부터 서명자에 이르기까지의 련쇄를 구성하는데 충분한 공개열쇠증명서들의 모임을 포함한다. 이 정보는 다음 base64로 부호화된다. 본보기통보문(RFC 822머리부를 제외하고)은 다음과 같다.

Content-Type: application/pkcs7-mime: smime-type=signed-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment;filename=smime.p7m

567GHIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
77n8HHGT9HG4VqpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH
HuujhJh4VqpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
6YT64VOGhIGfHfQbnj75

서명된 통보문을 회복하고 서명을 검증하기 위하여 수신측은 먼저 base64부호화를 해제한다. 다음 서명자의 공개열쇠로 요약통보문을 복호한다. 수신자는 요약통보문을 따로 처리하고 그것을 복호된 요약통보문과 비교하여 서명을 검증한다.

명백한 서명작성

명백한 서명작성은 서명된 보조부분형을 가지는 다중부분내용부형을 리용하여 달성된다. 언급한바와 같이 이 서명과정은 통보문이 "in the clear"로 보내지도록 서명되는 통보문을 변환하지 않는다. 이리하여 MIME능력은 있으나 S/MIME능력이 없는 수신자들이 받은 통보문을 읽을수 있다.

다중부분/서명통보문은 부분으로 되여 있다. 첫 부분은 임의의 MIME행일수 있는데 원천지로부터 목적지까지 전송될 때 교체되지 않도록 준비되여야 한다. 이것은 첫 부분이 7bit가 아니면 base64나 quoted-printable로 부호화되여야 한다는것을 의미한다. 또한 이 부분은 서명자료와 같은 방법으로 처리되는데 이 경우에 서명자료형식을 가진 객체가 창조된다. 그것은 빈 통보문내용부마당을 가진다. 이 객체는 독립서명이다. 다음 다중부분/서명통보문의 두번째 부분은 base64을 리용하여 부호화되는것이다. 이 두번째 부분은 응용의 MIME내용부형과 pkcs7서명의 부분형을 가진다. 아래에 본보기통보문을 소개하였다.

Content-Type: multipart/signed;
  Protocol= "application/pkcs7-signature";
  Micalg=sha1;boundary=boundary42

--boundary42
content-Type:text/plain

This is a clear — signed message.

```
--boundary42
content-Type:application/pkcs7-signature;name=smime,p7s
content-Transfer-Encoding:base64
content-Disposition:attachment;filename=smime.p7s

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VqpfyF467GhIGfHfYT6
4VqpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GHIGfHfYT64VQbnj756
--boundary42--
```

규약파라메터는 이것이 두 부분의 명백한 서명실체라는것을 가리킨다. Micalg파라메터는 리용된 요약통보문의 형을 지적한다. 수신자는 첫 부분의 요약통보문을 취하고 이것을 두번째 부분의 서명에서 재현한 요약통보문과 비교하는 방법으로 서명을 검증한다.

등록요구

일반적으로 응용프로그람이나 사용자는 공개열쇠증명서용의 증명국을 적용할수 있다. Application/pkcs10s/MIME실체는 증명요구를 전송하는데 리용된다.

증명요구는 증명요구정보블로크다음에 공개열쇠암호화알고리듬의 식별자 그 다음의 수신자비밀열쇠로 작성된 증명요구정보블로크서명으로 구성된다. 증명요구정보(Request Info)블로크는 증명서주제(공개열쇠가 확인되는 실체)의 이름과 사용자의 공개열쇠의 비트렬표현을 포함한다.

증명서전용통보문

증명서 또는 증명서취소목록(CRL)만을 포함하는 통보문은 등록요구에 응하여 보내질수 있다. 통보문은 smime형퇴화파라메터를 가진 application/pkcs7-mime type/subtype이다. 그 단계는 통보문내용이 없고 서명자정보마당이 빈다는것을 내놓고는 서명자료통보문의 창조과정과 같다.

S/MIME증명서처리

S/MIME는 X.509(11장을 볼것)의 판본 3과 일치하는 공개열쇠증명서들을 리용한다.

S/MIME가 리용하는 열쇠관리방식은 엄격한 X.509증명계층구조와 신용PGP의 Web사이의 혼합이다. PGP모형과 마찬가지로 S/MIME관리자들과 사용자 혹은 그 개개들은 매 의뢰자들을 신용되는 열쇠들의 목록과 증명서취소목록으로 구성하여야 한다. 즉 책임은 들어 온 서명들을 검증하고 내보낼 통보문들을 암호화하는데 필요한 증명서들을 유지하는데 국한된다. 다른 한편 증명서들은 증명국에 의해 서명된다.

사용자대행체의 역할(User Agent Role)

S/MIME사용자는 다음의 조작을 실현하기 위한 몇가지 열쇠관리기능들을 가진다.

- 열쇠생성: 일정한 행정사업을 보는 사용자(실례로 LAN관리와 관련한것)는 디

피-헬만(Diffie-Hellman)과 DSS열쇠쌍들 그리고 RSA열쇠쌍들을 생성할수 있다. 매 열쇠쌍은 비결정식우연입력으로부터 생성되여야 하며 안전하게 보호되여야 한다. 사용자대행체는 길이 768~1024-bitRSA열쇠쌍들을 생성해야 하며 512bit보다 작은 길이는 생성하지 말아야 한다.

- 등록:사용자의 공개열쇠는 X.509공개열쇠증명서를 받기 위해 증명국에 보관되여야 한다.
- 증명서보관과 검색: 사용자는 증명서들의 국부목록에 대한 접근을 요구하며 들어 온 서명들을 검증하고 내보낼 통보문들을 암호화한다. 이러한 목록은 많은 사용자들의 이름으로 사용자나 일부 제한된 행정실체에 의해 유지될수 있다.

Verisign Certificates

증명국(CA)봉사들을 제공하는 몇개의 회사들이 있다. 실례로 Nortel은 CA를 계획하였고 기관내에서 S/MIME지원을 제공할수 있다. Verisign, GTE 및 U.S.Postal Service를 비롯하여 수많은 인터네트 CA들이 있다. 이것들중에서 대부분이 VerisignCA봉사인데 그에 대해 간단히 설명하면 다음과 같다.

표 12-8. Verisign공개열쇠증명서급분류

	신원확인의 개요	IA비밀열쇠보호	증명서신청자 및 예약자의 비밀열쇠 보호	사용자들에 의해 실장되거나 예상되는 응용들
1급	명확한 이름과 전자우편주소의 자동탐색	PCA:신용되는 장치 CA: 신용되는 프로그람 또는 장치	요구되지 않지만 권고되는 암호프로그람(PIN보호)	Web열람과 전자우편리용
2급	1급과 같은데 자동명부정보검사와 자동주소검사가 더 있다.	PCA와 CA: 믿을수 있는 장치	암호프로그람(PIN보호)의 요구	개인 및 인트라네트와 회사들사이의 전자우편,직결가입,통과어교체 및 프로그람유효성검증
3급	1급과 같은데 개인대면과 ID문서들 그리고 2급의 개인들에 대한 즉 기관들의 업무기록(혹은 서류)들에 대한 자동ID검사가 더 있다.	PCA와 CA: 믿을수 있는 장치	암호프로그람 (PIN보호)요구: 장치교환권이 권고되지만 요구되지 않는다.	전자은행,회사자료기지접근, 개인은행, 종업원기초의 직결봉사, 내용완정성검사, 전자상업봉사기, 프로그람유효성검증, LRAAs의 인증; 어떤 봉사기들에 대하여 강한 암호화

IA: 발행국
CA: 증명국
PCA: 확인서명 공개주증명국
PIN: 개인식별번호
LRAA: 국부등록주관리자

Verisign은 S/MIME나 다른 여러 응용프로그람들에 량립될수 있도록 CA봉사를 제공한다. Verisign은 제품명이 Verisign Digital ID인 X.509증명서들을 발행한다. 1998년초에 35000개이상의 상업Web싸이트들이 Verisign server Digital ID들을 리용하였으며 100만개이상의 고객수자식식별자(DID)들이 Netscape나 Microsoft열람기들의 사용자들에게 발행되였다.

- 소유자의 공개열쇠
- 소유자의 이름이나 가명
- Digital ID의 완료날자
- Digital ID의 계렬번호
- Digital ID를 발행한 증명국의 이름
- Digital ID를 발행한 증명국의 수자서명

수자식ID들에는 또한 다른 사용자제공정보도 포함될수 있다.실례로

- 주소
- 전자우편주소
- 기본등록정보(나라, zip부호, 나이, 성)

Verisign은 공개열쇠증명서보안의 세개의 준위 또는 급부류들을 제공한다. 사용자는 Verisign의 Web싸이트나 다른 web싸이트들에서 증명서를 직렬로 요구한다. 1부류와 2부류요구들을 직결로 처리되며 대부분의 경우 증명하는데 몇초 걸린다. 간단히 다음의 절차들이 리용된다.

수자식ID에 포함된 정보는 수자식ID와 그 사용의 형태에 의존한다. 최소한 매개 수자식ID는 다음의것을 포함한다.

- 1급수자식ID에서 Verisign은 PIN과 수자식ID발취정보를 응용에서 제공된 전자우편주소에 보내여 사용자의 전자우편주소를 확인한다.
- 2급수자식ID에서 Verisign은 1급의 수자식ID와 관련한 모든 검사를 수행하는것외에 고객자료기지와의 자동비교를 통하여 접수된 정보를 검증한다. 마지막으로 수자식ID가 그의 이름으로 발행된 사용자를 바꾼 특정의 우편주소로 확인이 보내진다.
- 3급수자식ID에서 Verisign은 더 높은 준위의 신원증명을 요구한다. 개별적사람들은 자기의 신원을 신임장을 제공하거나 본인이 직접 신청하여 증명해야 한다.

개선된 보안봉사(Enhanced Security Services)

현재까지 3가지 개선된 보안봉사들이 인터네트초안으로 제기되였다. 그 구체적내용은 달라 질수도 있고 부가적인 봉사들이 더 있을수도 있다. 그러한 봉사들은 다음과 같다.

- **서명된 수신:** 서명된 수신 혹은 서명된 자료객체로부터 요구될수 있다. 서명된 수신자에로의 회답은 통보문의 작성자에로의 배달에 대한 증명을 제공하고 그

작성자로 하여금 수신자가 그 통보문을 받았다는것을 제3자가 증명할수 있게 한다. 본질적으로 수신자는 서명자의 서명이 있는 원래의 통보문에 서명하고 새로운 S/MIME통보문을 구성하는 새 서명을 덧단다.

- **보안표식:** 보안표식은 서명된 자료객체의 인증된 특성에 포함될수 있다. 보안표시는 S/MIME교갑화에 의해 보호되는 내용의 민감성에 관한 보안정보들의 모임이다. 그 표시들은 어떤 사용자가 객체에 대해 접근허가를 받는가를 지적하는 접근조종에 리용할수 있다. 다른 사용들에는 어떤 사람들이 그 정보를 볼수 있는가를 서술하는 우선권이 포함된다.
- **안전우편목록:** 사용자가 다중수신자들에게 통보문을 보낼 때 매 수신당 일정한 처리량이 요구된다. 사용자는 S/MIME우편목록대행체(MLA)의 봉사를 고용하여 이 작업에서 면제될수 있다. MLA는 받은 개개의 통보문을 취하고 매개 수신에 대하여 수신자전용암호화를 진행하고 그 통보문을 전송할수 있다. 그 통보문의 작성자는 MLA의 공개열쇠로 암호화를 진행하여 MLA에 그 통보문을 보내는것만 요구한다.

참고문헌

참고할 Web싸이트들

- PGP Home Page: PGP Web site by Network Associates, the leading PGP commercial vendor
- MIT Distribution Site for PGP: Leading distributer of free ware PGP contains FAQ. Other information and links to other PGPsites
- S/MIME Charter: Latest RFCs and internet drafts for S/MIME
- S/MIME Central: RSAInc.'s Web site for S/MIME . Includes FAQ and other useful information

문 제

1. 대부분의 전통암호들에서 암호블로크련쇄(CBC)방식을 리용하지만 PGP는 CAST-128의 암호반결합(CBC)방식을 리용한다.

$$\text{CBC: } C_i = E_k[C_{i-1} \oplus P_i];\ P_i = C_{i-1} \oplus D_k[C_i]$$
$$\text{CFB: } C_i = P_i \oplus E_k[C_{i-1}];\ P_i = C_{i-1} \oplus E_k[C_{i-1}]$$

   이 두개는 동등한 보안을 제공하는것으로 되고 있다. PGP가 CFB방식을 리용하는 리유를 설명하시오.
2. PGP방식에서 이전에 창조된 대화열쇠가 만들어 지기전에 생성될수 있는 대화열쇠들의 수는 얼마인가?
3. PGP에서 N개의 공개열쇠들을 가진 사용자가 최소 한개의 중복된 열쇠ID를 가

질 확률은 얼마인가?

4. PGP서명에서 128-bit통보문의 요약정보의 첫 16bit들이 명백히 분석된다.
   ① 이것은 어느 정도까지 하쉬알고리듬의 보안을 낮추는가?
   ② 그것은 의도된 기능 즉 정확한 RSA열쇠가 리용되여 요약정보가 복호되였는가를 결정하려는 기능을 어느 정도까지 수행하는가?
5. 그림 12-4에서 공개열쇠고리의 매개 입력은 이 공개열쇠소유자와 관련된 신용의 정도를 가리키는 소유자신용마당을 포함한다. 그것은 왜 충분하지 않은가? 즉 소유자가 신용되고 그것이 소유자의 공개열쇠라고 가정해도 왜 PGP가 이 공개열쇠을 리용하는것을 허락하는데 충분한 신용으로 되지 못하는가?
6. 암호의 형태로서 radix-64변환을 고찰하자. 이 경우 열쇠는 없다. 그러나 적이 어떤 형태의 치환알고리듬이 영어본문을 암호화하는데 리용되였다는것만을 알았다고 가정하자. 이 알고리듬은 암호분석을 막는데서 얼마나 효과적이겠는가?
7. 힐 집머맨(Phil Zimmermann)은 IDEA, 3중DES 및 CAST-128을 PGP를 위한 규격암호알고리듬으로 선택하였다. 이 책에서 서술된 다른 매개 전통암호알고리듬들 즉 DES, 두개열쇠3중DES, Blowfish, RC2과 RC5가 PGP에 적당하거나 그렇지 못한 리유를 밝히시오.

부록 12-1: ZIP를 리용한 자료압축

PGP는 진 루프 가일리(Jean-lup Gailly), 마크 애들러(Mark Adler) 및 리챠드 웨일스(Richard wales)가 쓴 이른바 ZIP라는 압축묶음을 리용한다. ZIP는 UNIX와 일부 다른 체계들에서 실행되는 C언어로 작성된 자유품(freeware)묶음이다. ZIP는 기능적으로 PKWARE회사에 의해 개발된 Windows체계들에 유용한 공유품(shareware)묶음인 PKZIP와 같다. Zip알고리듬은 가동환경들에 관계없이 가장 일반적으로 쓰이는 압축기술이며 freeware와 shareware판본들은 Windows나 UNIX체계들과 마찬가지로 Macintosh와 다른 체계들에서 응용된다.

ZIP와 류사한 알고리듬들이 야꼬브지브(JacobZiv)와 아브라함 렘펠(Abraham Lempel)에 의하여 개발되였다. 1977년에 벌써 그들은 최근에야 처리된 본문[ZIV77]이 있는 미끄럼창문완충기에 기초한 기술을 서술하였다.

이 알고리듬을 일반적으로 LZ77이라고 부르는데 그 판본은 zip압축방식에 리용된다(PKZIP, gzip, zipit 등).

LZ77과 그의 변종들은 본문흐름(GIF의 경우 화상패턴들)에서 단어들과 구들이 반복될수 있다는 사실을 리용하였다. 반복부들이 발생할 때 반복되는 렬은 짧은 부호로 교체될수 있다. 압축프로그람은 이러한 반복을 조사하고 반복되는 계렬을 교체하기 위하여 실행중에 부호들을 개발한다. 후에 부호들이 새로운 계렬들을 포착하는데 재리용된다. 이 알고리듬은 압축풀기프로그람이 부호들과 원천자료계렬들사이의 현행의 넘기기를 추론할수 있도록 정의되여야 한다.

LZ77를 구체적으로 보기전에 간단한 실례를 보자. 무의미한 구

the brown fox jumped over the brawn foxy jumping frog

를 고찰하자. 이것은 53옥테드 즉 424-bit길이이다. 알고리듬은 이 본문을 왼쪽에서 오른쪽으로 처리해 간다. 처음 매 문자는 2진 1과 그 다음 그 문자의 8-bitASCII표현으로 이루어 지는 9-bit의 패턴으로 넘기기된다. 처리가 진행될 때 알고리듬은 반복되는 계렬들을 찾는다. 반복부들과 만나게 되면 알고리듬은 그 반복이 끝날 때까지 계속 조사한다. 다시말하여 반복이 발생할 때마다 알고리듬은 될수록 많은 문자들을 포함한다. 이러한 첫 렬은 the brown fox이다.

이 렬은 선행계렬에 대한 지적자나 그 렬의 길이로 교체된다. 이 경우에 the brown fox의 앞렬은 26개 문자위치들이 앞에 발생하며 그 렬의 길이는 13문자들이다. 이 실례에서 부호화하는데 두가지 선택 즉 8-bit지시기와 4-bit길이 또는 12-bit지시기와 6-bit길이를 가정한다. 2bit의 머리부는 어떤 선택이 되는가를 가리킨다. 00는 첫 선택을 가리키고 01은 두번째 선택을 가리킨다. 따라서 the brown fox의 두번째 발생은 $<00_b><26_d><13_d>$ 또는 00 00011 010 1101로 부호화된다.

압축된 통보문의 나머지부분은 문자 Y, 공백문자와 그 다음의 jump로 이루어 지는 렬에 교체되는 렬 $<00_b>$ $<26_d>$ $<5_d>$, 문자렬 ing frog이다.

그림 12-9에 압축넘기기를 보여 주었다. 압축된 통보문은 35개의 9-bit문자들과 두개의 부호들을 포함하여 총 35×9+2×14=343bit들로 이루어 진다. 이것은 압축하지 않은 통보문의 424bit에 비해 1.24배의 압축률을 가진다.

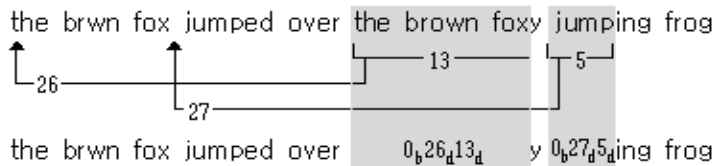



그림 12-9. LZ77방식의 실례

압축알고리듬

LZ77과 그 변종들에 대한 압축알고리듬은 두개의 완충기들을 리용한다. 미끄럼리력완충기(Siding history buffer)는 처리된 원천의 마지막 N개 문자들을 포함하고 전망완충기(look-ahead buffer)는 처리되여야 할 다음 L개 문자들을 포함한다(그림 12-10의 ㄱ).

알고리듬은 look-ahead buffer의 시작으로부터 sliding history buffer에 있는 기호렬까지 대조하려고 한다. 만일 일치되는것이 없으면 look-ahead buffer의 첫 문자는 9-bit문자로서의 출구이고 sliding window에로 밀기화되면서 그안에서 제일 낡은 문자는 밀기화된다.

만일 일치되는것이 발견되면 알고리듬은 제일 긴 일치가 나올 때까지 계속 조사한다. 그러면 일치된 기호렬은 세 조(지적자, 지시기, 길이)로서 출력된다.

K-문자렬에 대하여 sliding window의 *K*개의 제일 오랜 문자들은 밀기되고 부호화된 기호렬의 *K*개의 문자들은 window에로 밀기된다.

그림 12-10의 ㄴ에는 우의 실례렬에서 이 방식의 동작을 보여 주었다. 그림에서 39-문자 미끄럼창문과 13-문자 미리보기완충기를 가정한다. 이 실례의 웃부분에서 첫 40개

문자들이 처리되고 이 문자들중의 가장 최근의 39개의 압축되지 않는 판본이 sliding window에 있다. 나머지원천은 look-ahead window에 있다. 압축알고리듬은 다음일치를 결정하고 look-ahead buffer로부터 5개의 문자들을 sliding window에로 밀기하고 이 기호렬에 대한 부호를 출력한다. 이런 조작들후 완충기의 상태는 아래의 그림과 같다.

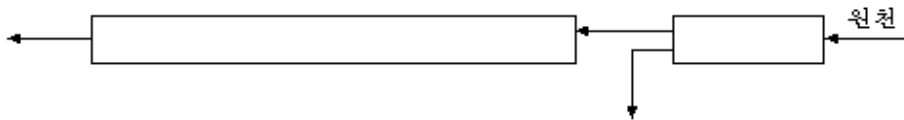
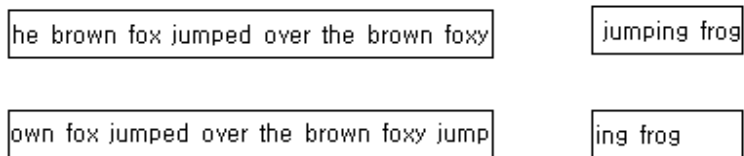



그림 12-10. LZ77도식

LZ77이 효과적이고 현재입력의 특성에 부합되지만 일부 부족점들이 있다. 이 알고리듬은 앞의 본문에서 일치되는것들을 찾는데 유한창문을 리용한다. 창문의 크기에 비해 매우 긴 블로크의 본문에서는 가능한 많은 일치들이 제거될수 있다. 창문크기를 증가시킬수 있는데 이때 두개의 벌칙들이 부가된다. 즉 (1) sliding window의 모든 위치에 대하여 look-ahead buffer의 렬비교를 해야 하므로 알고리듬의 처리시간이 증가한다. (2) <pointer>마당은 큰 비약에 대해서는 지적자마당이 커야 한다.

압축풀기알고리듬

LZ77-압축된 본문의 풀기는 간단하다. 압축풀기알고리듬은 풀기된 출력의 마지막 N개의 문자들을 보존하여야 한다. 부호화된 기호렬에 부닥칠 때 압축풀기알고리듬은 <pointer>와 <length>마당들을 리용하여 그 부호를 실지 본문기호렬로 교체한다.

부록 12-2: 64-진수변환

PGP와 S/MIME는 둘다 **radix-64변환**이라고 부르는 부호화기술을 리용한다. 이 기술은 임의의 2진자료를 인쇄가능한 문자출력으로 넘긴다. 부호화형식의 특성들은 다음과 같다.

1. 함수의 범위는 모든 싸이트들에서 보편적으로 표현할수 있는 문자들의 모임이며 그 문자모임은 특정한 2진부호화는 아니다. 따라서 문자들자체는 특정한 체계에 필요한 어떤 형식으로 부호화된다. 실례로 문자 "E"는 ASCⅡ체계에서 16진45로 표시되며 EBCDIC체계에서는 16진C5로 표시된다.
2. 문자모임은 65개의 인쇄가능한 문자들로 이루어 지는데 그중 하나는 삽입에 쓰인다. 2^6=64개의 유용한 문자중에서 매개 문자들은 6bit의 입력을 표현하는데 리용된다.
3. 조종문자들은 그 모임에 속하지 않는다. 따라서 radix-64로 부호화된 통보문은 조종문자들의 자료흐름을 조사하는 우편조종체계들을 관통할수 있다.
4. 이음표 《-》는 리용되지 않는다. 이 문자는 RFC 822형식에서 의미를 가지며 따라서 표 12-9는 6-bit입력값들의 문자들에로의 넘기기를 보여 준다. 문자모임은 문자, 수자들과 《+》 및 《/》로 이루어 진다. 《=》문자는 삽입문자로 리용한다.

그림 12-11에 간단한 넘기기방식을 보여 주었다. 2진입력은 3개의 8bit들의 블로크들로 처리된다. 24-bit블로크에서 매 6bit들의 모임은 한개 문자로 넘기기된다. 그림에서 문자들은 8-bit량들로 부호화된것으로 볼수 있다. 이 일반경우에 매 24-bit입력은 32-bit출력으로 확장된다.

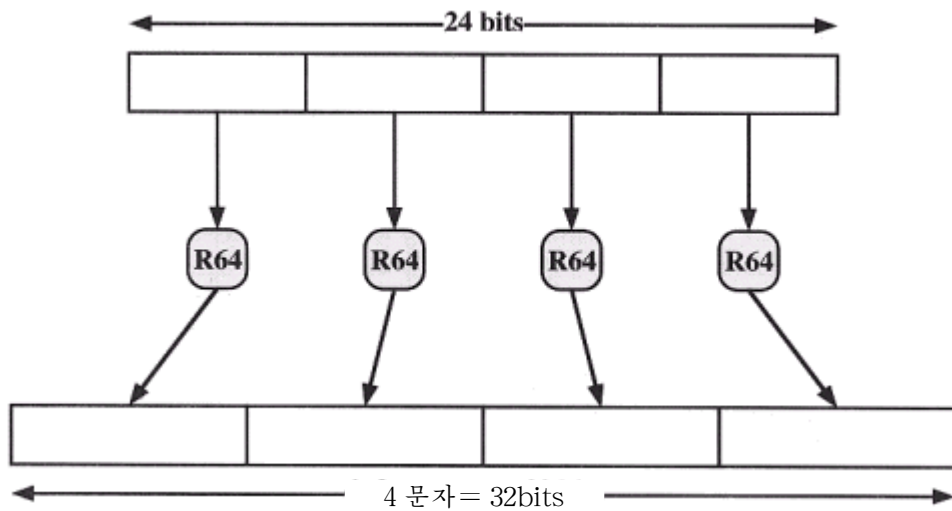



그림 12-11. 2진자료의 Radix-64형식으로서 인쇄가능한 부호화

실례로 16진으로 235C91로 표시할수 있는 24bit 미가공본문렬 00100011 01011100 10010001을 고찰하자. 그 입력을 6-bit블로크들로 배렬한다.

001000 110101 110010 010001

발취된 6-bit10진값들은 8,53,50,17이다. 이것들을 표 12-9에서 찾아 보면 다음의 문자들 즉 I1yR로서 radix-64부호화를 준다. 만일 기우성비트를 0으로 설정하고 이 문자들을 8-bitASCⅡ형식으로 보관하면 01001001 00110001 01111001 01010010을 얻는다.

16진수로 이것은 49317952이다. 개괄하면

입력자료	
2진표현	00100011 01011100 10010001
16진표현	235C91
입력자료의 Radix-64부호화	
문자표현	I1yR
ASCⅡ부호(8bit, 령, 기우성)	01001001 00110001 01111001 01010010
16진표현	49317952

표 12-9. Radix-64부호화

6-bit값	문자부호화	6-bit값	문자부호화	6-bit값	문자부호화	6-bit값	문자부호화
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(pad)	=

부록 12-3: PGP우연수생성

PGP는 여러 목적으로 우연수들과 모조란수들을 생성하는데 복잡하고 유력한 방식을 리용한다. PGP는 우연수들을 내용부와 사용자건눌림의 시간결정으로 생성하고 모조란수들도 ANSI X12.17의 하나에 기초한 알고리듬을 리용하여 생성한다. PGP는 다음의 목적들을 위하여 이 수들을 생성한다.

- 진짜수
  - RSA열쇠쌍들을 생성하는데 쓰인다.
  - 모조란수생성의 초기씨를 제공
  - 모조란수생성기간 첨부입력을 제공
- 모조란수
  - 대화열쇠들을 생성하는데 리용된다.
  - CFB방식의 암호에서 대화열쇠를 리용하기 위한 초기화벡토르(IV)들을 생성하는데 쓰인다.

진짜우연수

PGP는 256-byte완충기의 우연비트들을 유지한다. PGP가 건눌림을 제외할 때마다 그 시간을 대기를 시작한 32-bit형식으로 기록한다. PGP가 건눌림을 받을 때 건이 눌린 시간과 그 건눌림의 8-bit값을 기록한다. 그 시간과 건눌림정보는 열쇠를 생성하는데 리용되며 열쇠는 또 우연-비트완충기의 현재값을 암호화하는데 리용된다.

모조란수

모조란수발생은 24-octet씨를 리용하여 16-octet대화열쇠, 8-octet초기화벡토르 그리고 모조란수발생에 리용될 새로운 씨를 만든다. 알고리듬도 5장에서 서술한 X12.17 알고리듬에 기초하는데(그림 5-14를 볼것) 암호화에는 3중DES대신 CAST-128을 리용한다.

1. 입력
   - randseed.bin(24악테드): 이 파일이 비면 24개의 진짜 우연악테드들로 채워 진다.
   - 통보문: 대화열쇠와 통보문을 암호화하는데 리용될 초기화벡토르들은 그 통보문의 함수이다. 이것은 또한 열쇠와 초기화벡토르의 우연성에 보다 기여되며 적이 통보문의 평문내용부를 이미 알고 있을 때 1회용대화열쇠를 획득할 필요가 분명치 않게 한다.
2. 출력
   - K(24개 악테드): 첫 16개 악테드 K[0, ...15]은 대화열쇠를 포함하며 마지막 8개 악테드 k[16, ...23]은 초기화벡토르를 포함한다.
   - randseed.bin(24개 악테드):이 파일에는 새로운 씨값이 놓인다.

3. 내부자료구조

- dtbnf(8개 옥테드): 첫 4개의 옥테드 dtbnf[0,...3]은 현재날자/시간값으로 초기화된다. 이 완충기는 X12.17알고리듬의 DT변수와 같다.
- rkey(16개 옥테드):알고리듬의 모든 단계들에서 리용되는 CAST-128암호열쇠
- rseed(8개 옥테드):X12.17의 V_i변수와 같다.
- rbuf(8개 옥테드):알고리듬에 의해 생성된 모조란수. 이 완충기는 X12.17의 R_i변수와 같다.
- K'(24개 옥테드): randseed.bin의 새로운 값을 위한 림시완충기

알고리듬은 G1부터 G9까지의 9개 단계들로 이루어 진다. 처음과 마지막단계들은 적에 도청당한 randseed.bin파일의 값을 줄이기 위한 장애단계들이다. 나머지 단계들은 X12.17알고리듬의 세개의 반복들과 본질적으로 같은데 그림 12-12에 보여 주었다(그림 5-14와 비교하시오).

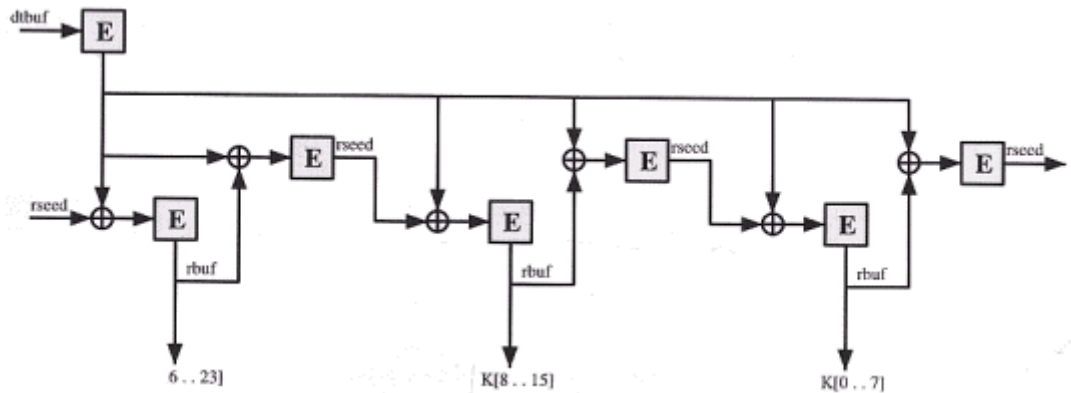



그림 12-12. PGP대화열쇠와 IV생성(단계 G2에서 G8까지)

알고리듬의 다음과 같은 단계마다의 서술은 Stephan Nenhaus[NEUH93]에 기초한다.

G1. [초보적으로 이전의 씨를 처리한다.]

ㄱ) randseed.bin을 k[0..23]까지 복사하시오.

ㄴ) 그 통보문의 하쉬를 취한다(이것은 통보문이 서명되면 미리 생성되고 그렇지 않으면 통보문의 첫 4K옥테드들이 리용된다). 그 결과를 열쇠로 리용하며 빈 초기화벡토르를 리용하여 K를 CFB방식으로 암호화하고 그 결과를 K에 다시 보관한다.

G2. [초기씨를 설정한다]

ㄱ) dtbuf[0..3]을 32-bit국부시간으로 설정한다. dtbnf[4..7]을 모두 령으로 설정한다. rkey ← k[0..15]를 복사한다. rseed←K[16..23]를 복사한다.

ㄴ) 64-bit의 dtbuf를 128-bit의 rkey를 리용하여 FCB방식으로 암호화하고 그 결과를 dtbuf에 다시 보관한다.

G3. **[우연옥테드들을 생성하기 위한 준비]**rcount←0 k←23을 설정한다. 단계 G4-G7의 순환은 24번(k=23..0) 실행되며 매번 한개의 옥테드가 생성되여 K에 보관된다. 변수 rcount는 rbuf에서 리용되지 않는 우연옥테드들의 수이다. 8부터 0까지 3번 내리계수하여 24개의 옥테드들을 생성한다.

G4. **[바이트들이 유용한가?]** rcount=0이면 G5로 가고 아니면 G7로 간다. G5나 G6은 X12.17알고리듬의 한개 단계를 수행하여 8개의 우연옥테드들의 새로운 묶음을 생성한다.

G5. **[새로운 우연옥테드들을 생성한다]**

ㄱ) rseed←rseed⊕dtbuf

ㄴ) rbnf←Erkdy[rseed] (ECB방식으로)

G6. **[다음씨를 생성한다.]**

ㄱ) rseed← rbnf⊕dtbuf

ㄴ) rseed← Erkey⊕[rseed] (ECB방식으로)

ㄷ) rcount← 8

G7. **[rbuf로부터 K에로 한번에 한개 바이트를 전송한다.]**

ㄱ) rcount←rcount-1

ㄴ) 진짜우연바이트 b를 생성하고 k[k] ←rbuf[rcount]⊕b를 설정한다.

G8. **[다 되였는가?]** K＝0이면 G9로 가고 아니면 k← k-1 및 G4로 간다.

G9. **[씨를 Postwash하고 그 결과를 돌려 준다]**

ㄱ) G7에서 우연바이트로 XOR하지 않는것을 내놓고 G4-G7의 방법으로 24개 이상의 바이트들을 생성한다. 그 결과를 완충기 K'에 보관한다.

ㄴ) K'를 열쇠 k[0..15]와 초기화벡토르 K[16..23]을 리용하여 CFB방식으로 암호화하고 그 결과를 randseed.bin에 보관한다.

ㄷ) K를 돌려 준다.

G12.a에서 생성된 24개의 새 옥테드들로부터 대화열쇠를 결정할수는 없다. 그러나 기억한 randseed.bin파일이 최근의 대화열쇠들에 대한 어떠한 정보도 제공하지 않도록 하기 위하여 24개의 새로운 옥테드들을 암호화하고 그 결과를 새로운 씨로 보관한다.

이 정교한 알고리듬은 암호학적으로 강한 모조란수들을 제공하게 된다.

알고리듬의 예비적분석에 의해 단일대화열쇠의 비트들사이에는 내적련관이 없으며 다음의 대화열쇠들도 독립이라는것을 알수 있다[NEUH 93].

제13장. IP보안

인터네트통신은 전자우편(S/MIME, PGP), 의뢰기/봉사기(Kerberos), Web호출(Secure Sockets Layer) 등을 비롯하여 많은 응용령역들에서 응용프로그람고유의 보안수단들을 개발하였다. 그러나 사용자들에게는 규약층들을 위반하는 일부 보안상우려들이 있다. 실례로 어떤 회사는 안전한 전용TCP/IP망을 의심되는 싸이트들에 련결하지 않고 구내에서 나가는 파케트들을 암호화하며 구내에 들어 오는 파케트들을 인증하는 방식으로 운영할수 있다. IP수준에서 보안을 완성하여 기관은 보안대책을 취한 응용뿐아니라 보안을 무시한 많은 응용들에 대하여서도 담보할수 있다.

IP수준의 보안은 세개의 기능적령역들인 인증, 기밀성 및 열쇠관리를 포함한다. 인증기구는 수신된 파케트가 파케트머리부에서 발송지로 지적된 대방에 의해 전송되였다는 것을 확인한다. 또한 이 기구는 그 파케트가 전송중에 변경되지 않았다는것을 담보한다. 기밀성은 제3자에 의한 도청을 막는다. 열쇠관리는 안전한 열쇠의 교환과 관련된다.

13.1 IP보안에 대한 개괄

1994년에 인터네트구성방식위원회(Internet Architecture Board:IAB)는 "Security in the Internet Architecture"라는 제목의 보고를 발표하였다(RFC 1636). 그 보고는 인테네트가 더 많은 그리고 더 훌륭한 보안을 요구한다는 일반여론을 지적하고 보안기구들에 대하여 기본령역들을 확인하였다. 그중에서 허가 받지 못한 감시나 망전송의 조종으로부터 망의 하부구조를, 인증과 암호기구들로는 말단 대 말단사용자전송을 보호할 필요가 있다.

이 우려들에는 충분한 근거들이 있다. 콤퓨터긴급대책팀(Computer Emergency Response Team:CERT)의 1997년도 보고에서는 거의 150,000개의 싸이트들에 영향을 주는 2500개이상의 보안련관사건들을 지적하였다 .대부분 엄중한 형태의 공격들은 IP기만의 형태를 띠는데 여기서 침입자들은 거짓 IP주소를 가진 파케트들을 창조하고 IP에 기초한 인증을 리용하는 응용들 즉 여러 형태의 도청과 파케트탐지로 가입정보와 자료기지내용들을 비롯한 정보를 알아 낸다.

이에 대응하여 IAB는 다음세대 IP에서 필요한 보안대책으로서 인증과 암호화를 포함하는 IPv6을 발행하였다. 이것은 판매업자들이 이 특성들을 제공할수 있다는것을 의미하며 현재 많은 판매업자들이 자기들의 상품에 IPSec능력을 갖추고 있다.

IPSec의 응용

IPSec는 LAN, 전용 및 광지역망(WAN)들 그리고 인터네트를 통한 통신을 보호하는 능력을 제공한다. 그 리용실례는 다음과 같다.

- **인터네트를 통한 지점에로의 통신을 보호한다:** 회사는 인터네트 또는 WAN에서 안전한 가상전용망을 구축할수 있다. 이것은 업무가 인터네트에 의존함으로써 전용망들에 대한 요구를 줄이고 비용과 망관리간접비를 절약할수 있게 한다.

- **인테네트에서 안전한 원격접근:** 체계가 IP보안규약들로 장비된 말단사용자는 인테네트봉사제공자(Internet Service Provider:ISP)에 국부접근하여 회사망에 안전하게 접근할수 있다.이것은 려행중에 있는 직원들의 원격작업의 비용을 줄인다.
- **엑스트라네트와 인트라네트에 대방들을 접속시킨다:** IPSec는 인증과 기밀성,열쇠교환수단들을 제공하며 다른 기관들과의 안전한 통신에 리용할수 있다.
- **전자상거래의 보안강화:**일부 Web와 전자상거래응용프로그람들이 보안수단들을 가지고 있다고 해도 IPSec를 리용하면 그 보안이 강화된다.

IPSec가 이러한 응응들을 지원할수 있게 하는 기본특성은 그것이 IP준위에서 모든 거래를 암호화하거나 인증할수 있다는것이다. 따라서 원격등록가입, 의뢰기/봉사기, 전자우편, 파일전송, 접근을 비롯한 모든 분산응용들이 담보된다.

그림 13-1에 IPSec사용법에 대한 전형적씨나리오를 보여 주었다. 기관들은 분산된 지역들에서 LAN들을 관리한다. 불안전한 IP통신이 매개 LAN에서 진행된다. Trafficoff싸이트에 대하여 어떤 전용 또는 공공WAN을 통하여 IPSec규약들이 리용된다. 이 규약들을 경로조종기나 방화벽과 같은 망장치들에 적용한다.

IPSec망장치는 WAN에로 가는 모든 전송들을 일반적으로 암호화하고 압축하며 WAN으로부터 오는 전송을 복호하고 압축을 푼다. 즉 이 조작들은 워크스테이션이나 봉사기들에 대하여 투명하다. 또한 안전한 전송이 WAN에 직접 접근하는 개별적사용자들에게도 가능하다. 이러한 사용자워크스테이션들은 IPSec규약들을 실행하여 보안을 보장한다.

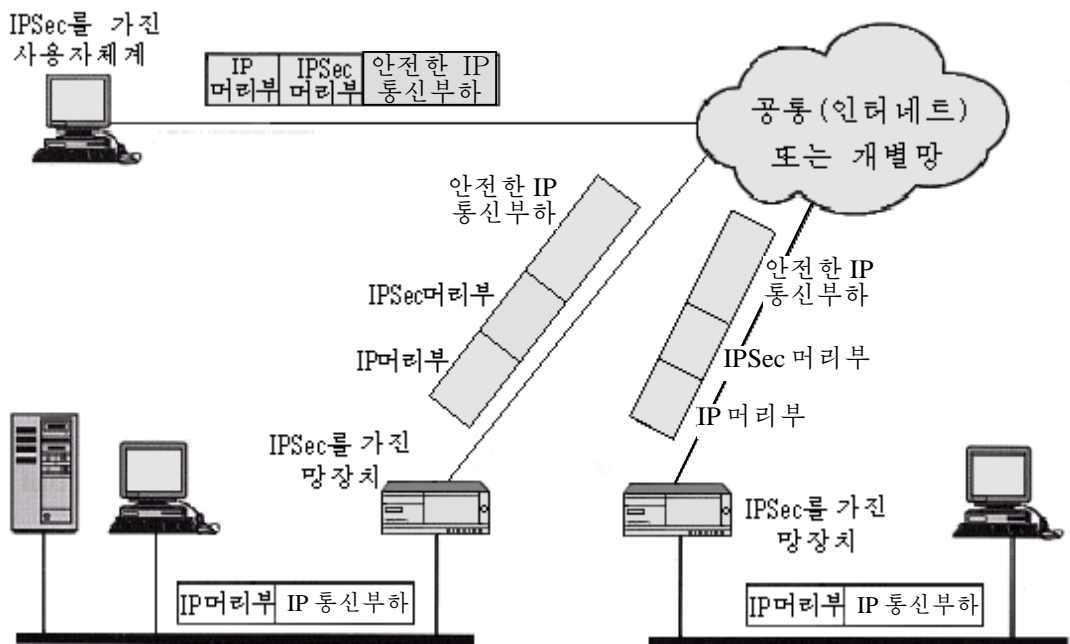



그림 13-1. IP보안의 씨나리오

IPSec의 우점

IPSec의 우점들은 다음과 같다([MARK97]).

- IPSec가 방화벽이나 경로조종기에 실장되면 그것은 경계를 지나는 모든 전송들에 적용할수 있는 강한 보안을 제공한다. 회사나 연구그룹내에서의 전송은 보안련관처리의 간접비를 초래하지 않는다.
- 외부로부터의 모든 전송이 IP를 리용하고 방화벽이 인터네트에서 그 기관으로 들어 오는 유일한 수단이면 방화벽에서 IPSec는 우회를 막게 된다.
- IPSec는 전송층(TCP, UDP)밑에서 실현되고 따라서 응용프로그람들에 투명하다. IPSec가 방화벽이나 경로조종기에서 실현될 때 사용자나 봉사기체계에 쏘프트웨어를 갈아 태울 필요는 없다. IPSec가 비록 말단체계들에서 수행될지라도 응용프로그람들을 비롯하여 웃층의 쏘프트웨어들에 영향을 주지 않는다.
- IPSec는 말단사용자들에 대하여 투명하다.사용자들이 보안기구들에 숙련할 필요가 없다.
- IPSec는 필요하다면 개별적사용자들에게 보안을 제공할수 있다. 이것은 원격지(off-site)의 작업자들과 기관내에 안전한 가상부분망을 설정하는데 유용하다.

경로선택응용

말단사용자들을 지원하고 구내체계들과 망들을 보호하는것외에 IPSec는 호상련결망에서 요구되는 경로선택방식에서 사활적인 역할을 논다. 문헌 [HUIT98]에 IPSec의 리용에 대한 다음의 실례들이 소개되였다.

- 경로조종기통지(새로운 경로기는 자기의 존재를 통지한다.)는 공인된 경로조종기에서 진행한다.
- 이웃한 통지(경로조종기는 다른 경로선택령역의 경로조종기와의 이웃한 관계를 창설하거나 유지하려고 한다.)는 공인된 경로조종기로부터 한다.
- 재방향통보문은 초기파케트가 보내온 경로조종기로부터 만든다.
- 경로선택변경은 꾸며 낼수 없다.

이러한 보안대책이 없으면 적은 통신을 와해시키거나 일부 전송의 방향을 바꿀수 있다. OSPF와 같은 경로조종규약들은 IPSec에 의해 정의된 경로조종기들사이의 보안련관상에서 실행된다.

13.2 IP보안구성방식

IPSec명세서(specification)는 보다 복잡해 졌다. 전반적방식에 대한 표상을 가지기 위해 IPSec를 정의하는 문서들을 보는것으로부터 시작한다. 다음 IPsec와 봉사기들을 론의하고 보안련관(security association)의 개념을 소개한다.

IPSec문서

1995년 8월에 IETF는 인터네트수준에서 보안을 정의하는 다섯개의 보안에 관련하여 제안된 표준들을 발행하였다.

- RFC 1825: 보안방식에 대한 륜곽
- RFC 1826: IP에 대한 파케트인증확장의 서술
- RFC 1828: 명확한 인증방식
- RFC 1827: IP에 대한 파케트암호화확장의 서술
- RFC 1829: 명확한 암호방식

이 특성들에 대한 지원은 IPv6에서는 의무적이며 IPv4에서는 자유이다. 두 경우 다 보안기능들은 기본IP머리부에 련이은 확장머리부들로서 실현된다. 인증을 위한 확장머리부를 인증머리부라고 한다. 또한 암호화에 대한 확장머리부는 교갑화보안통신부하(Encapsulating Security Payload: ESP)머리부라고 한다.

이 첫 문서들로부터 IETF에 의해 설립된 IP보안규약연구그룹(IP Security Protocol Working Group)에서 많은 연구사업들이 진행되고 있다. 문서들은 그림 13-2에서 서술된것처럼 7개의 부류들로 분할된다.

- **방식:** 일반개념들, 보안조건들 정의들 및 IPSec기술을 정의하는 방식들을 포함한다.
- **교갑화보안통신부하(ESP):** 파케트암호화를 위한 ESP의 리용과 관련한 파케트형식과 일반결과들, 선택적기능인 인증을 포함한다.

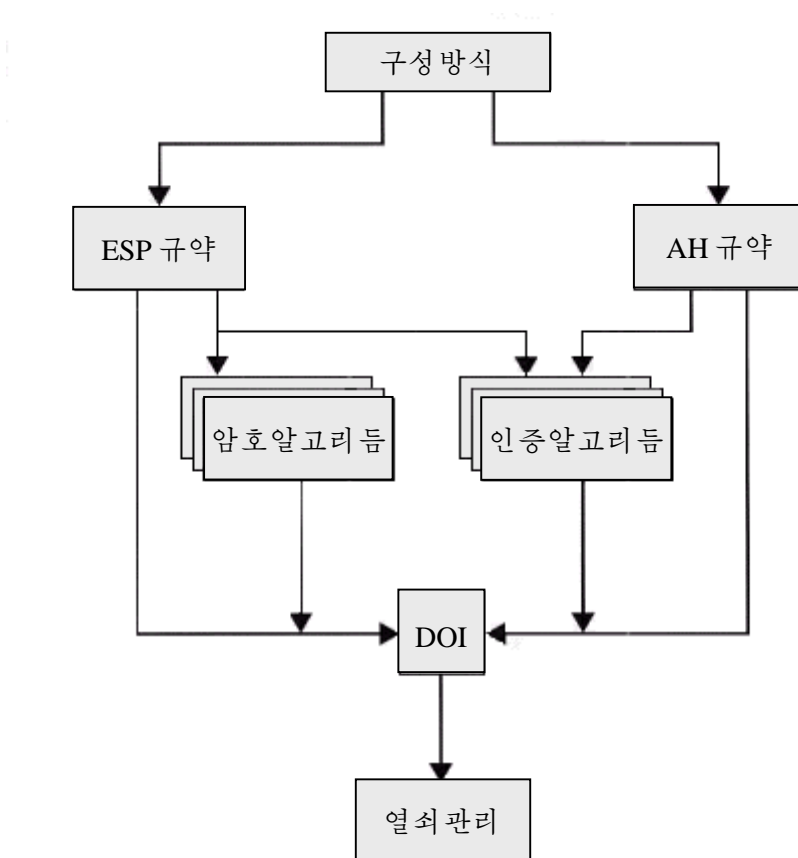



그림 13-2. IPSec문서개괄

- **인증머리부(AH):** 파케트인증을 위한 AH의 리용과 관련한 파케트형식과 일반결과들을 포함한다.
- **암호알고리듬:** 어떤 암호알고리듬들이 ESP를 위해 리용되는가를 서술하는 문서들의 모임
- **인증알고리듬:** 어떤 인증알고리듬들이 AH와 ESP의 인증선택에 리용되는가를 서술하는 문서들의 모임
- **열쇠관리:** 열쇠관리수단들을 서술하는 문서들
- **해석령역(DOI):** 다른 문서들이 서로 관계를 가지는데 필요되는 값들을 포함한다. 이것들은 열쇠의 수명과 같은 조작파라메터들과 마찬가지로 공인된 암호 및 인증알고리듬들에 대한 식별자들을 포함한다.

IPSec봉사

IPSec는 체계가 요구하는 보안규약들을 선정하고 그 봉사들을 위해 리용할 알고리듬들을 결정하며 요청 받은 봉사를 제공하는데 요구되는 모든 암호열쇠들을 준비할수 있게 하여 IP층에서 보안봉사들을 제공한다. 보안을 제공하는데 두개의 규약이 리용된다. 규약의 머리부에 의해 지정되는 인증규약인 AH와 그 규약에 대한 파케트의 형식에 의해 지정되는 암호화/인증규약인 ESP가 있다.

봉사들은 다음과 같다.

- 접근조종
- 비접속완정성의 보증
- 자료출처의 인증
- 재연된 파케트들(부분렬완정성의 형식)의 거부
- 기밀성(암호화)
- 제한된 전송흐름기밀성

표 13-1에 어떤 봉사가 AH와 ESP규약에 의해 제공되는가를 보여 주었다. ESP에 대하여 두가지 경우가 있다. 즉 인증선택을 가지는것과 가지지 않는것이 있다. AH와 ESP는 둘 다 암호열쇠배포와 이 보안규약들에 관계되는 전송흐름의 관리에 기초한 접근조종을 위한 운반수단들이다.

표 13-1. IPSec봉사

	AH	ESP(암호만)	ESP(인증과 암호화)
접근조종	✓	✓	✓
비접속완정성	✓		✓
자료출처의 인증	✓		✓
재연된 파케트들의 거부	✓	✓	✓
기밀성		✓	✓
제한된 전송흐름의 기밀성		✓	✓

보안련관(Security Associations)

IP에 대하여 인증과 기밀성구조에서의 중요한 개념이 보안련관(Security Association :SA)이다. 관련(association)은 그 우에서 진행되는 전송에 보안봉사를 제공하는 송신자와 수신자사이의 한방향관계이다. 쌍방향의 안전한 교환에서 동등한 관계를 요구하면 두개의 보안련관들이 요구된다. 보안봉사들이 AH 또는 ESP의 리용을 위해 SA에 제공되는데 둘다에 대하여서는 아니다.

SA(Security association)는 다음의 세 파라메터들에 의해 유일하게 정의된다.

- **보안파라메터목록**(Security Parameter Index:SPI): 이 SA에 할당된 비트렬이며 국부적의미만을 가진다. SPI는 AH와 ESP머리부들에 옮겨 져 수신체계가 수신되는 파케트를 처리할 SA를 선택할수 있게 한다.
- **IP목적지주소**（IP Destination Address）: 현재의 unicast주소들만이 허락된다. 이것은 SA의 목적지 끝점의 주소인데 마지막사용자체계이거나 방화벽 또는 경로조종기와 같은 망체계일수 있다.
- **보안규약식별자**(Security Protocol Identifier): 이것은 관련이 AH보안련관인가 또는 ESP보안련관인가를 가리킨다.

따라서 임의의 IP파케트에서 SA는 IPv4 또는 IPv6머리부에서 목적지주소와 닫긴 확장머리부(AH나 ESP)의 SPI에 의해 유일하게 식별된다.

SA파라메터

매개의 IPSec실현에는 매 SA와 관련한 파라메터들을 정의하는 보안련관자료기지（Security Association Database）가 있다. SA는 일반적으로 다음의 파라메터들로 정의된다.

- **렬번호계수기**（Sequence Number Counter）: 3절에서 서술되는 AH나 ESP머리부들에 있는 Sequence마당을 생성하는데 쓰이는 32-bit값(모든 실현에서 요구된다).
- **계수기넘침**（Counter Overfolw）: 렬번호계수기의 자리넘침이 일어 났을 때 이 SA로 파케트들의 이후의 전송을 정지시키겠는가 정지시키지 않겠는가를 가리키는 기발(모든 실현들에서 요구된다).
- **반재연**(Anti-Replay): 3절에서 서술되는데 들어 오는 AH나 ESP파케트가 재연인가 아닌가를 결정하는데 쓰인다(모든 실현들에 요구된다).
- **AH정보**(AH Information): 인증알고리듬, 열쇠, 열쇠의 수명, AH를 리용하는것과 관련되는 파라메터들(AH실현에만 요구된다).
- **ESP정보**(ESP Information): 암호화 및 인증알고리듬, 열쇠초기값들, 열쇠의 수명 및 ESP를 리용하는데 관련되는 파라메터들(ESP실현에 요구된다).
- **현재보안련관의 수명**(Lifetime of this Security Association): 한 SA가 새로운 SA(및 새로운 SPI)로 교체되거나 끝 마치게 되는 시간구간 또는 바이트계수와 이 동작들중 어느것이 일어 날것이라는 지적(모든 실현들에 요구된다).
- **IPSec규약방식**(IPSec Protocol Mode): 턴넬, 전송 또는 윌드카드(모든 실현들에서 요구된다). 이 방식들은 이 절의 뒤에서 론의한다.
- **경로 MTU**: 임의의 관측된 경로최대전송단위(쪼각화를 하지 않고 전송할수 있

는 파케트의 최대크기) 및 경과시간을 나타내는 변수들(모든 실현들에서 요구된다).

열쇠를 배포하는데 쓰이는 열쇠관리기구는 보안파라메터목록의 방법으로만 인증과 비밀성꾸밈새들에 결합된다. 따라서 인증과 비밀성은 어떤 특정한 열쇠관리꾸밈새와 독립으로 기입된다.

SA선택기

IPSec는 사용자들에게 IPSec봉사들을 IP전송에 적용시키는 방법으로 상당한 유연성을 제공한다. SA들은 여러가지 방법으로 결합되여 소망하는 사용자구성을 만들수 있다. 또한 IPSec는 IPSec보호가 제공되는 전송과 IPSec를 우회할수 있는 전송을 어렵지 않게 판별한다. 전자의 경우에 IP전송은 특정의 SA들에 관련된다.

IP전송이 특정의 SA들(또는 IPSec를 우회하는것이 허락되는 전송의 경우에는 SA가 없다)에 관계되는 방법은 명목상의 보안확인절차자료기지(Security Policy Data base:SPD)이다. 그것의 가장 간단한 형태로서 SPD는 기입들을 포함하는데 그 매개는 IP전송과 부분설정을 정의하며 그 전송을 위한 SA를 지적한다. 더 복잡한 환경들에서는 잠재적으로 단일SA에 관련하거나 단일SPD기입과 관련되는 다중SA들과 관련한 다중기입들이 있을수 있다. 독자들은 충분한 리해를 위하여 IPSec관련문서들을 참고할수 있다.

매 SPD기입은 IP와 선택기라고 부르는 층의 규약마당의 값들의 모임에 의해 정의된다. 사실 이 선택기들은 나가는 전송을 특정의 SA에로 넘기기 위하여 그것을 려과하는데 리용된다. 외부에로의 처리는 매개 IP파케트에 대하여 다음의 일반적차례로 진행한다.

1. SPD에 대하여 파케트의 적당한 마당들의 값을 비교하여 일치하는 SPD기입을 찾는데 그것은 령 또는 그이상의 SA들을 지적한다.
2. 이 파케트와 그와 관련된 SPI에 대한것이 있으면 SA를 결정한다.
3. 요구된 IPSec처리를 한다(즉 AH나 ESP처리).

다음의 선택기들은 SPD기입을 결정한다.

- **목적지 IP주소(Destination ID):** 이것은 단일IP주소, 하나하나 렬거한 주소들의 목록이나 범위 또는 주소일수 있다. 마지막 두가지는 같은 SA를 공유한 한개이상의 목적지체계를 지원하는데 요구된다(실례로 방화벽뒤에).
- **원천지 IP주소(Source ID):** 이것은 단일IP주소, 하나하나 렬거된 주소목록이나 범위 또는 윌드카드(마스크)주소일수 있다. 마지막 두가지는 같은 SA를 공유하는 하나이상의 원천국체계를 지원하는데 요구된다.
- **사용자식별자(UserID):** 조작체계로부터의 사용자의 식별자. 이것은 IP나 웃층의 머리부들의 마당은 아니지만 만일 IPSec가 사용자와 같은 조작체계에서 실행되면 리용가능하다.
- **자료감도준위(Data Sensitivity Level):** 정보의 흐름보안을 제공하는 체계들에서 리용된다.
- **IPSec규약(AH나 ESP 또는 AH/ESP):** 만일 존재하면 이것은 IPv4규약이나

IPv6 의 Next Header마당으로부터 얻어 진다.

- **전송층규약(Transport Layer Protocol):** IPv4규약이나 IPv6 Next Header 마당으로부터 얻어 진다. 이것은 개별적규약번호, 규약번호들의 목록 또는 규약번호들의 범위일수 있다.
- **원천지 및 목적지포구들(Source and Destination Ports):** 이것들은 개별적 TCP나 UDP포구값들, 상세히 렬거된 포구목록 또는 윌드카드포구일수 있다.
- **IPv6클라스(IPv6 Class):** IPv6머리부에서 얻어 진다. 이것은 개별적IPv6클라스값이나 윌드카드값일수 있다.
- **Ipv6흐름표식(IPv6 Flow Label):** IPv6머리부에서 얻어 진다. 이것은 개별적 IPv6표식값이나 윌드카드값이 될수 있다.
- **Ipv4봉사형(Type of Service:TOS):** IPv4머리부에서 얻어 진다. 이것은 개별적IPv4의 TOS값이나 윌드카드값이 될수 있다.

전송방식과 턴넬방식

AH와 ESP는 둘다 두가지 방식 즉 전송과 턴넬방식을 지원한다. 이 두가지 방식의 조작에 대해서는 AH와 ESP의 해설을 통해 충분히 리해할수 있는데 그에 대해 3절과 4절에서 서술하였다. 여기서는 간단한 개괄만을 준다.

전송방식

전송방식(transport mode)은 웃층의 규약들에 대하여 초보적으로 보호를 제공한다. 즉 전송방식의 보호는 IP파케트의 통신부하까지 미친다. 실례들에 TCP, UDP토막 또는 인터네트조종통보문규약(Internet Control Message Protocol:ICMP)파케트가 포함되는데 그모두는 주콤퓨터규약모임(stack)에서 IP에 대하여 직접 조작된다. 일반적으로 전송방식은 두개의 주콤퓨터들사이의 말단 대 말단통신에 리용된다(즉 의뢰기와 봉사기, 또는 두개의 워크스테이션). 주콤퓨터가 IPv4우에서 AH나 ESP를 리용할 때 통신부하은 일반적으로 IP머리부에 따르는 자료이다. IPv6에 대하여 통신부하은 보호에 포함될수 있는 목적지선택머리부를 제외하고는 IP머리부와 존재하는 임의의 IPv6확장머리부들에 따르는 자료이다.

전송방식에서 ESP는 IP통신부하을 암호화하고 선택적으로 인증도 하지만 IP머리부에 대하여서는 그렇게 못한다. 전송방식에서 AH는 IP통신부하과 IP머리부의 선택된 부분들을 인증한다.

턴넬방식

턴넬방식(tunnel mode)은 전체 IP파케트에 대한 보호를 제공한다. 이것을 결정하기 위해 AH나 ESP마당들이 IP파케트에 첨부된 다음 보안마당들에 전체 파케트를 합치여 새로운 《외부(outer)》IP머리부를 가지는 새로운 외부IP파케트의 통신부하으로서 취급한다. 전체 초기의 또는 내부(inner)파케트는 턴넬을 통하여 IP망의 한 곳에서 다른 곳으로 옮겨 간다. 즉 경로의 도중에 있는 경로조종기는 내부파케트를 볼수 없다. 처음의 파케트가 교갑화되여 있으므로 새로운 더 큰 파케트는 보안에 대하여 총체적으로 다른 원천지와 목적지주소들을 가진다. 턴넬방식은 SA의 하나 또는 두개의 말단들이 IPSec를 실현하는 방화벽이나 경로조종기와 같은 보안관문일 때 리용된다. 턴넬방식에서 방화벽들뒤의 많은 가입자들은 IPSec를 실현하지 않고 안전한 통신에 들어 갈수 있

다. 이 가입자들에 의해 생성된 보호되지 않은 파케트들은 국부망의 경계에서 방화벽이나 안전한 경로조종기의 IPSec쏘프트웨어에 의해 설정된 턴넬방식 SA들에 의하여 외부망들을 통하여 턴넬화된다.

여기에 턴넬방식의 IPSec가 어떻게 동작하는가를 보여 주는 실례가 있다. 망우의 가입자 A는 다른 망의 가입자 B의 목적지주소를 가지는 IP파케트를 생성한다. 이 파케트는 A가 속한 망경계에서 초기가입자로부터 방화벽이나 안전한 경로조종기에로 경로가 지정된다. 방화벽은 모든 나가는 파케트들을 려파하여 IPSec처리의 요구를 결정한다. 만일 A로부터 B에 가는 파케트가 IPSec를 요구하면 방화벽은 IPSec처리를 하고 그 파케트를 외부IP머리부로 교갑한다. 이 외부IP파케트의 원천지IP주소는 그 방화벽으로 되며 목적지주소는 B의 국부망의 경계를 형성하는 방화벽이 된다. 이 파케트는 B의 방화벽에서 외부IP머리부들만을 조사한다. B의 방화벽에서 외부IP머리부는 벗겨 지며 내부파케트는 B에게 배포된다.

턴넬방식에서 ESP는 내부IP머리부를 포함하여 전체 내부파케트를 암호화하고 선택적으로 인증한다. 턴넬방식의 AH는 전체 내부IP파케트와 외부IP머리부의 선택된 부분들을 인증한다. 표 13-2에 전송 및 턴넬방식의 기능을 개괄하였다.

표 13-2. 턴넬방식과 전송방식의 기능

	전송방식 SA	**턴넬방식 SA**
AH	IP통신부하과 IP머리부의 선택된 부분들 및 IP∨6확장머리부들을 인증한다.	전체 내부IP파케트와 외부IP머리부의 선택된 부분들을 합치여 인증하고 외부IP∨6확장머리부들을 인증한다.
ESP	ESP머리부에 따라 IP통신부하과 모든 IP∨6확장머리부들을 암호화한다.	내부IP파케트를 암호화한다.
인증을 가지는 ESP	ESP머리부에 따라 IP통신부하과 모든 IP∨6확장머리부들을 암호화한다. IP머리부는 내놓고 IP통신부하을 인증한다.	내부IP파케트를 암호화한다. 내부IP파케트를 인증한다.

13.3 인증머리부

인증머리부(Authentication Header)는 IP파케트들의 인증과 자료의 완정성에 대한 지원을 제공한다. 자료의 완정성특성은 전송중에 파케트내용의 변경을 적발할수 있게 하는것이다. 인증의 특징은 말단체계나 망장치가 사용자나 응용프로그람을 인증하고 따라서 전송을 려파할수 있게 한다. 또한 오늘날 인터네트에서 자주 볼수 있는 주소기만공격을 방지할수 있다. 그리고 AH는 재연공격도 막는다.

인증은 8장에서 서술한것처럼 통보문인증코드(message authentication code: MAC)에 기초하며 따라서 그 두 대방들은 비밀열쇠를 공유해야 한다.

인증머리부는 다음의 마당들로 이루어 진다(그림 13-3).

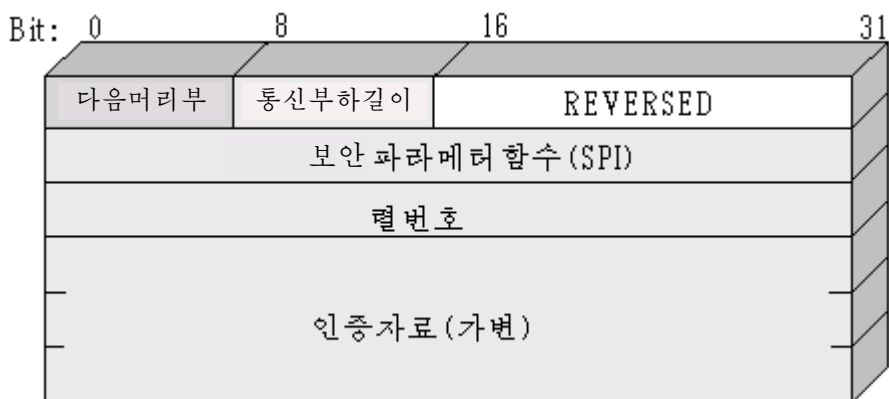



그림 13-3. IPSec인증머리부

- Next Header(8bit): 이 머리부에 련이은 머리부의 형을 식별한다.
- Payload Length(8bit): 32-bit단어로서 인증머리부의 길이에 2를 던 값. 실례로 인증자료마당의 표준길이는 96bit 즉 3개의 32-bit단어이다. 3단어고정머리부와 함께 머리부에는 모두 6개의 단어들이 있는데 마당은 값 4를 가진다.
- Reversed(16bit): 앞으로의 리용을 위한것이다.
- Security Parameters Index(32bit): 보안련관을 확인한다.
- Sepuence Number(32bit): 단조적으로 증가하는 계수기값. 후에 론한다.
- Authentication Data(가변): 이 파케트에 대하여 완정성검사값(ICV)이나 MAC를 포함하는 변수길이마당(32-bit단어들의 옹근수여야 한다).

반재연봉사

재연공격은 공격자가 인증된 파케트를 복사하고 후에 그것을 공격대상으로 되는 목적지에 전송하는 공격이다. 인증된 IP파케트들을 재사용하는 수신측은 어떤 방법으로 봉사를 혼란시키거나 일부 다른 예상외의 결과를 초래할수 있다. Sequence Number마당은 이러한 공격들을 막기 위해 설계되였다. 먼저 송신자에 의한 렬번호발생을 론의한 다음 그것이 수신자에 의해 어떻게 처리되는가를 보자.

새로운 SA가 설정되면 송신자는 렬번호계수기를 0으로 초기화한다. 파케트가 이 SA에 보내질 때마다 송신자는 계수기를 증가시키고 그 값을 Sequence Number마당에 설정한다. 따라서 리용되는 첫 값은 1이다. 만일 재연공격을 막을수 있게 된다면 송신자는 렬번호가 $2^{32}-1$을 지나 다시 0으로 순환하지 않도록 해야 한다. 그렇지 않으면 같은 렬번호를 가지는 여러개의 정당한 파케트들이 있게 된다. 만일 $2^{32}-1$의 한계에 도달되면 송신자는 이 SA를 끝내고 새로운 열쇠를 가지고 새 SA를 교섭한다.

IP가 비접속(connectionless)이고 믿을수 없는 봉사이므로 규약은 파케트들이 순서대로 배렬되며 모든 파케트들이 배달된다는것을 담보하지 못한다. 따라서 IPSec 인증문서는 **수신자**가 W=64의 기정값을 가지는 크기 W의 창문을 실장하여야한다는것을 지적한다.

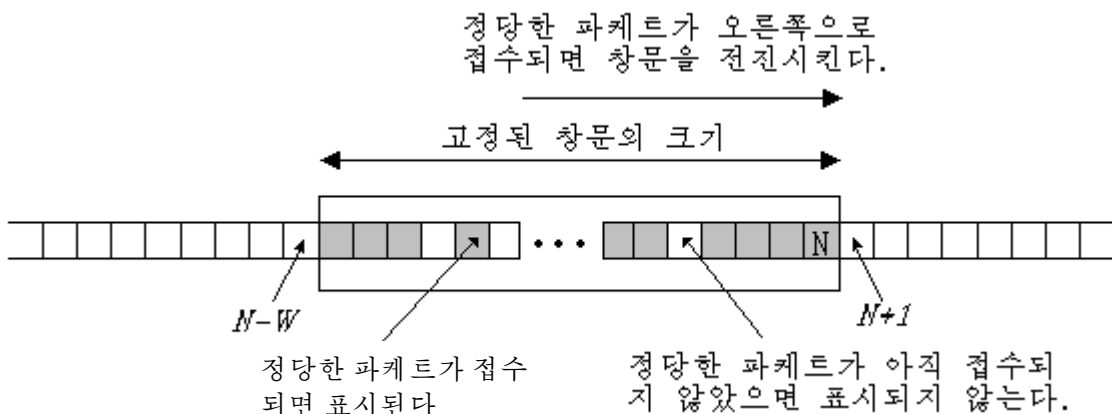



그림 13-4. 반재연꾸밈새

창문의 오른쪽 끝은 정당한 파케트를 접수했을 때까지의 제일 높은 렬번호 N을 표시한다. 렬번호가 $N-W+1$부터 N까지 범위에 있는 정확히 수신된 모든 파케트들에 대하여 창문에는 해당한 홈들이 표시된다(그림 13-4). 파케트를 접수하였을 때 돌아 오는 처리는 다음과 같이 진행된다.

1. 수신된 파케트가 창문안에 떨어 지고 새것이면 MAC가 검사된다. 만일 그 파케트가 확인되면 창문에서 대응하는 홈은 표시된다.
2. 만일 수신된 파케트가 창문의 오른쪽에 있고 새것이면 MAC를 검사한다. 그리고 그 파케트가 확인되면 창문은 이 렬번호가 창문의 오른쪽 끝이 되도록 전진하고 대응하는 홈은 표시된다.
3. 수신된 파케트가 창문의 왼쪽에 있거나 인증이 실패하면 그 파케트는 버린다. 즉 이것은 검사할수 있는 사건이다.

완정성검사값

인증자료마당은 완정성검사값(Integrity Check Value: ICV)이라고 부르는 값을 취한다. 완정성검사값은 통보문인증코드이거나 MAC알고리듬에 의해 만들어 진 코드를 생략한것이다. 현재의 명세서에 근거한 실현은 다음의 것을 지원해야 한다는것을 시사해준다.

- HMAC-MD5-96
- HMAC-SHA-1-96

이 두개는 다 HMAC알고리듬을 리용하는데 전자는 MD5하쉬코드를, 후자는 SHA-1하쉬코드를 리용한다(이 모든 알고리듬들은 9장에서 서술되였다). 두 경우에 충분한 HMAC값이 계산되는데 다음인증자료마당의 기정길이인 처음의 96bit로 요약하여 쓸수 있다.

MAC는 다음과 같이 계산된다.

- 전송중에 변하지 않거나 AHSA에 대하여 말단에 도착한 값으로 예측할수 있는 IP머리부마당들. 전송중에 변할수 있고 또 도착한 값이 예측 불가능한 마당들은 발송지와 목적지 두 곳에서 계산을 위하여 령으로 설정한다.
- AH머리부는 인증자료마당과 다르다. 인증자료마당은 발송지와 목적지에서 계산을 위하여 령으로 설정한다.
- 전송중에 변하지 않는다고 가정되는 전체적인 웃준위의 규약자료(실례로 턴넬방식에서 TCP토막이나 내부IP파케트).

IPv4에 대하여 변하지 않는 마당들의 실례들은 Internet Header Length와 Source Address이다. 예측 가능한 마당을 내놓고 변하는 마당의 실례는 Destination Address이다. ICV계산에 대한 우선권이 령으로 되는 변하는 마당들의 실례는 Time to Live와 Header Checksum마당들이다. 주소기만이 방지되도록 발송지와 목적지주소 마당들이 다 보호된다는데 주의해 둔다.

IPv6에 대하여 기본머리부에서의 실례들로는 Version(변경불가능), Destination Address(예측가능을 제외하고 변경가능) 및 Flow Label(추정에 대하여 변경 및 령으로 되는)이 있다.

전송 및 턴넬방식

그림 13-5에는 IPSec인증봉사를 리용할수 있는 두가지 방법을 보여 주었다. 한가지 경우로는 인증이 봉사기와 의뢰기워크스테이션에 직접 제공되는것이다. 즉 워크스테이션은 봉사기와 같은 망이나 외부망에 있을수 있다. 워크스테이션과 봉사기가 보호된 비밀열쇠를 공유하는 한 인증과정은 안전하다. 이 경우는 전송방식 SA를 리용한다. 다른 경우 원격의 워크스테이션은 전체 내부망에 대한 호출에 대해 요청 받은 봉사기가 인증특성을 지원하지 않기때문에 자기를 공동의 방화벽에 확인시킨다. 이 경우는 턴넬방식 SA를 리용한다.

이 소절에서는 AH에 의해 제공되는 인증의 범위와 두가지 방식들에서 인증머리부의 위치를 본다. IPv4와 IPv6에 대한 고찰방법에는 차이가 있다. 그림 13-6의 ㄱ에 전형적인 IPv4와 IPv6파케트들을 보여 주었다. 이 경우 IP통신부하부분은 TCP토막이다. 즉 UDP나 ICMP와 같은 IP를 리용하는 임의의 다른 규약에 대한 자료단위일수도 있다.

IPv4를 리용하는 전송방식 AH에 대하여 AH는 원래의 IP머리부다음과 IP통신부하전에 삽입된다. 이것은 그림 13-6의 ㄴ의 웃부분에서 보여 준다. 인증은 MAC계산을 위하여 령으로 설정된 IPv4머리부의 변경가능한 마당들을 제외하고 전체 파케트를 포함한다.

IPv6과 관련하여 AH는 말단 대 말단통신부하으로 볼수 있다. 즉 중개경로조종기들에 의하여 조사되거나 처리되지 않는다. 따라서 AH는 IPv6기초머리부와 중계점머리부, 경로조종머리부 및 쪼각확장머리부들다음에 나타난다. 목적지선택확장머리부는 요구되는 의미에 따라 AH머리부의 앞이나 뒤에 나타날수 있다. 인증은 MAC계산을 위하여 령으로 설정된 변경가능한 마당들을 제외하고 파케트전체를 포함한다.

턴넬방식 AH에 대하여 전체적인 본래의 IP파케트가 인증되고 그 AH는 원래의 IP머리부와 새로운 외부의 IP머리부(그림의 ㄷ)사이에 삽입된다. 내부IP머리부는 원천지와 목적지의 주소를 나르는데 이때 외부IP머리부는 서로 다른 IP주소들을 포함할수 있다(즉 방화벽이나 다른 보안관문들의 주소들).

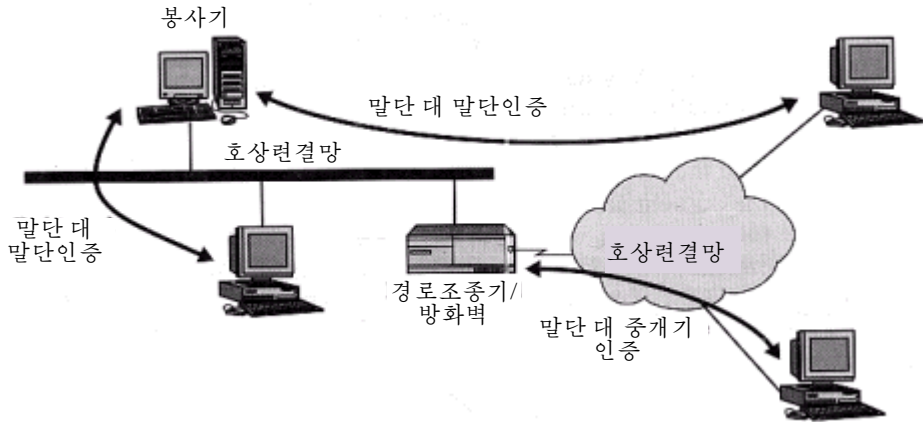



그림 13-5. 말단 대 중개자에 비한 말단 대 말단인증

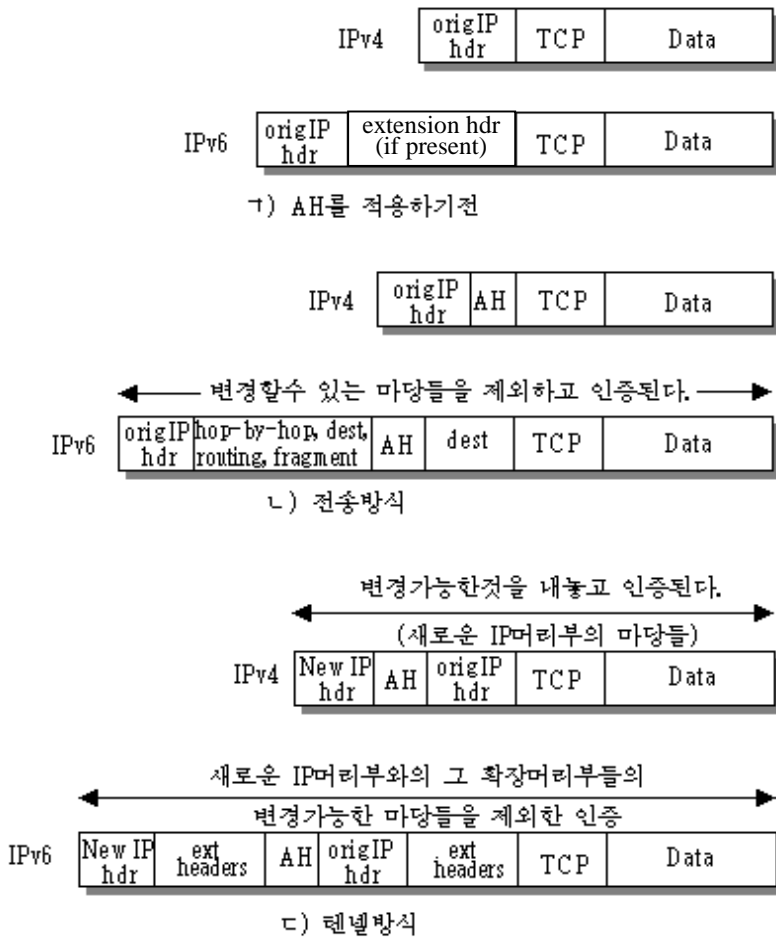



그림 13-6. AH인증의 범위

턴넬방식에서 전체 내부IP머리부를 포함하는 전체적인 IP파케트는 AH에 의해 보호된다. 외부IP머리부(그리고 IPv6의 경우에 외부IP확장머리부들)는 변경 및 예측불가능한 마당들을 제외하고는 보호된다.

13.4 보안통신부하의 교갑화

교갑화보안통신부하(Encapsulating Security Payload)는 통보문내용들의 기밀성과 전송흐름의 기밀성을 비롯한 기밀성봉사들을 제공한다. ESP는 선택적특성으로서 AH와 같은 인증봉사들도 제공할수 있다.

ESP형식

그림 13-7에 ESP의 파케트형식을 보여 주었다. 그것은 다음의 마당들을 포함한다.

- **Security Parameters Index(32bit):** 보안련관을 식별한다.

Sequence Number(32bit): 단조적으로 증가하는 계수기값. 이것은 AH에서 론의된것처럼 반재연기능을 제공한다.

- **Payload Data(가변):** 이것은 암호화에 의하여 보호되는 전송준위의 토막(전송방식) 또는 IP파케트(턴넬방식)이다.
- **Padding(0-255byte):** 이 마당의 목적은 후에 론의한다.
- **Pad Length(8bit):** 이 마당의 바로 앞의 메꾸기바이트수를 가리킨다.
- **Next Header(8bit):** 통신부하자료마당에 포함된 자료의 형을 그 통신부하의 첫 머리부를 확인하여 식별한다(실례로 IPv6의 확장머리부 또는 TCP와 같은 웃층의 규약).
- **Authentication Data(가변):** 파케트에서 계산된 완정성검사값에서 인증자료마당을 내놓은 가변길이마당(옹근수개의 32-bit단어들이다).

암호화 및 인증알고리듬

통신부하정보, Padding, Pad Length 및 Next Header마당들은 ESP봉사에 의해 암호화된다. 만일 통신부하을 암호화하는데 쓰이는 알고리듬이 초기화벡토르(IV) 등의 암호학적동기화자료를 요구하면 그 자료는 통신부하자료마당의 시작에서 얻을수 있다. 만일 그것이 포함되면 IV는 그것이 암호문의 존재부분이라고 해도 보통 암호화되지 않는다.

현재의 명세서에 근거한 실현은 암호블로크련쇄방식(CBC)(3장에서 서술)으로 DES를 지원할것을 요구한다. 많은 다른 알고리듬들은 DOI문서에서 식별자들을 할당하고 따라서 암호화에 쉽게 리용될수 있다. 이것들은 다음의것들을 포함한다.

- 3열쇠3중DES
- RC5
- IDEA
- 3열쇠3중IDEA
- CAST
- Blowfish

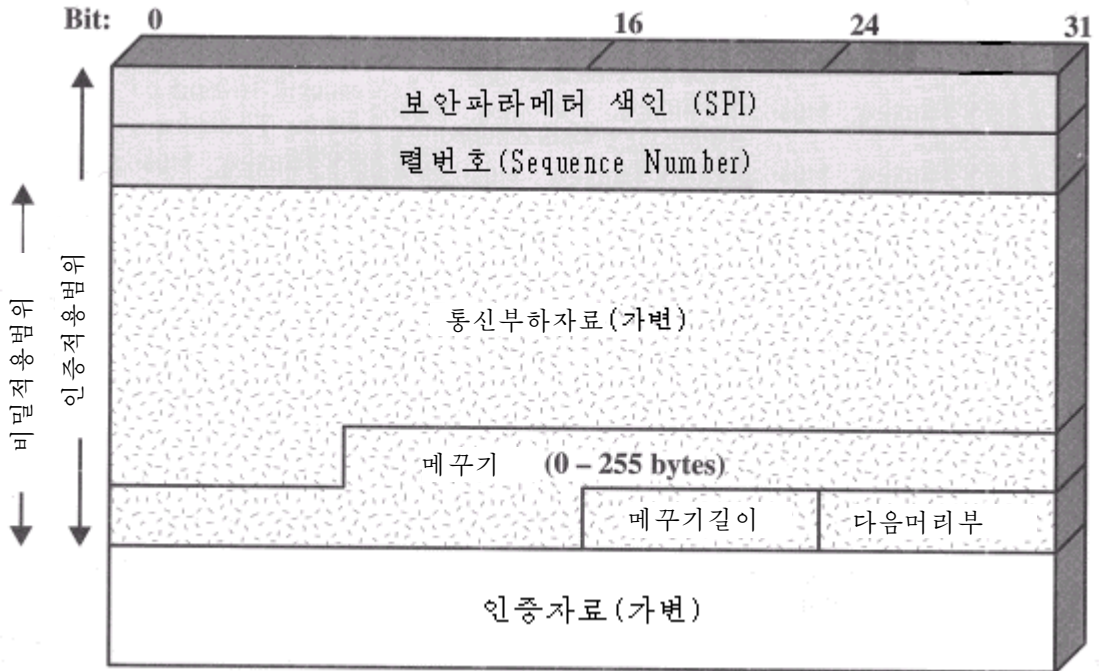



그림 13-7. IPSec ESP형식

이 알고리듬은 모두 4장에서 취급되였다.

AH와 마찬가지로 ESP는 96bit의 기존길이를 가지는 MAC의 리용을 지원한다. 또한 AH와 마찬가지로 현재의 명세서에 준한 실현을 위해서는 HMAC-MD5-96과 HMAC-SHA-1-96를 지원할것이 요구된다.

메꾸기

메꾸기마당은 다음의 몇가지 목적을 위하여 쓰인다.

- 만일 어떤 암호알고리듬이 몇개의 바이트들의 배수인 평문을 요구하면 Padding마당은 평문(Payload Data, Padding, Pad Length 및 Next Header마당들로 구성된)을 요구하는 길이까지 확장하는데 쓰인다.
- ESP형식은 Pad Length 및 Next Header마당들이 32-bit단어로 정렬될것을 요구한다. 마찬가지로 암호문은 32bit들의 옹근수배이여야 한다. Padding마당은 이 정렬을 보장하는데 리용한다.
- 부가적인 메꾸기는 통신부하의 실지길이를 비밀로 하여 부분적인 전송흐름의 기밀성을 제공하는데 첨부될수 있다.

전송 및 턴넬방식

그림 13-8에 IPSec봉사를 리용할수 있는 두가지 방법을 보여 주었다. 그림의 웃부

분에서 두 가입자들사이에 암호화를(선택적으로 인증) 직접 제공한다. 그림 13-8의 ㄴ에 턴넬방식의 조작을 리용하여 가상전용망(virtual private network)을 설정하는 방법을 보여 주었다.

이 실례에서 조작은 인터네트에 접속된 4개의 전용망을 가지고 있다. 내부망의 가입자들은 자료전송을 위하여 인터네트를 리용하지만 다른 인터네트기반의 가입자들과는 작용하지 않는다. 매개 내부망의 보안관문에서 턴넬들을 해방하면 그 구성은 가입자들의 보안기능의 수행을 중지하게 한다. 전자는 전송방식 SA에 의한 기술이고 후자는 턴넬방식의 SA를 리용하는 기술이다.

이 절에서는 두개의 방식들에 대한 ESP의 범위를 고찰한다. 고찰방법은 IPv4와 IPv6에서와는 좀 다르다. AH에 대한 론의와 마찬가지로 시작점으로서 그림 13-6의 ㄱ의 파케트형식들을 리용한다.

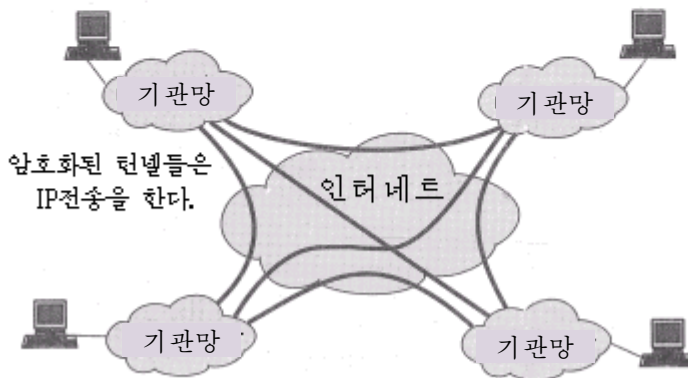
ㄱ) 전송준위보안

ㄴ) 턴넬방식을 경유한 가상사설망

그림 13-8. 전송방식 대 턴넬방식암호화

전송방식 ESP

전송방식 ESP는 그림 13-9의 ㄱ에서와 같이 IP에 의해 전송된 자료를 암호화하고 선택적으로 인증하는데 리용된다. IPv4를 리용하는 이 방식에 대하여 ESP머리부는 전

송층머리부(실례로 TCP, UDP, ICMP)의 바로앞의 IP파케트에 삽입되고 ESP끄리부(Padding, Pad Length 및 Next Header마당)가 IP파케트다음에 놓인다. 즉 인증이 선택되면 인증자료마당이 ESP끄리부다음에 부가된다. ESP끄리부에서 전송층의 토막들을 모두 합치여 암호화한다. 인증은 ESP머리부에 있는 모든 암호문들을 다 포함한다.

IPv6과 관련하여 ESP는 말단 대 말단통신부하으로 볼수 있다. 즉 ESP는 중개자경로조종기에 의하여 조사되거나 처리되지 않는다. 따라서 ESP머리부에서는 IPv6이 기초머리부와 중계점머리부, 경로조종머리부 및 쪼각확장머리부다음에 있게 된다.

목적지선택확장머리부는 요구에 따라 ESP의 전이나 후에 놓을수 있다. IPv6에 대하여 암호화는 만일 그것이 ESP머리부다음에 있으면 목적지선택확장머리부와 ESP끄리부,그리고 전체 전송층토막을 포함한다. 또한 인증은 ESP머리부에 암호문을 합하여 진행한다.

전송방식의 조작은 다음과 같이 개괄할수 있다.

1. 발송지에서 전체 전송층토막에 ESP끄리부를 합하여 이루어 지는 자료의 블로크가 암호화되고 그 블로크의 평문은 그것의 암호문으로 교체되여 전송을 위한 IP파케트로 형성된다. 인증선택이 되면 인증이 부가된다.
2. 다음 그 파케트는 목적지에로 경로선택된다. 매 중개경로조종기들은 모든 평문의 IP확장머리부들에 IP머리부를 합하여 조사하고 처리할것을 요구하지만 암호문은 조사하지 않는다.
3. 목적지마디는 임의의 평문IP확장머리부들에 IP머리부를 합하여 조사하고 처리한다. 다음 ESP머리부의 SPI에 기초하여 목적지마디는 평문전송층토막을 포함하는 그 파케트의 나머지를 복호한다.

전송방식의 조작은 그것을 리용하는 임의의 응용에 대하여 기밀성을 제공하며 따라서 모든 개별응용들에서 기밀성을 실현하지 않아도 된다. 또한 이 방식의 조작은 IP파케트의 전체길이에 조금 부가되므로 매우 효과적이다. 이 방식에서 한가지 약점은 전송된 파케트들에 대한 전송해석이 가능한것이다.

턴넬방식의 ESP

턴넬방식의 ESP는 전체 IP파케트를 암호화하는데 쓰인다(그림13-9의 ㄴ). 이 방식에서 머리부는 그 파케트의 앞에 놓이며 다음 ESP끄리부에 파케트를 더하여 암호화한다. 이 방법은 전송해석(traffic analysis)을 반격하는데 리용된다.

IP머리부가 목적지주소와 가능한 원천지경로조종명령 및 도약별 선택정보를 포함하므로 ESP머리부의 앞에 놓인 암호화된 IP파케트의 전송을 간단히 할수 없다.

중개경로조종기들은 이러한 파케트를 처리할수 없을것이다. 따라서 전체 블로크를 전송해석이 아니라 경로조종에 충분한 정보를 포함하는 새로운 IP머리부로 교갑화하여야한다. 여기서 전송방식은 ESP특성을 지원하는 가입자들사이의 련결을 보호하는데 알맞는 반면에 턴넬방식은 방화벽을 포함하거나 외부망으로부터 망을 보호하는 다른 종류의 보안관문이나 방화벽을 포함하는 구성에서 쓸모 있다. 후자의 경우에 암호화는 외부의 가입자와 보안관문 또는 두개의 보안관문들사이에서만 진행한다. 이것은 내부망의 가입자들에게서 암호화의 처리부담을 덜어 주며 요구되는 열쇠의 수를 줄여 열쇠배포문제를 간단하게 한다.

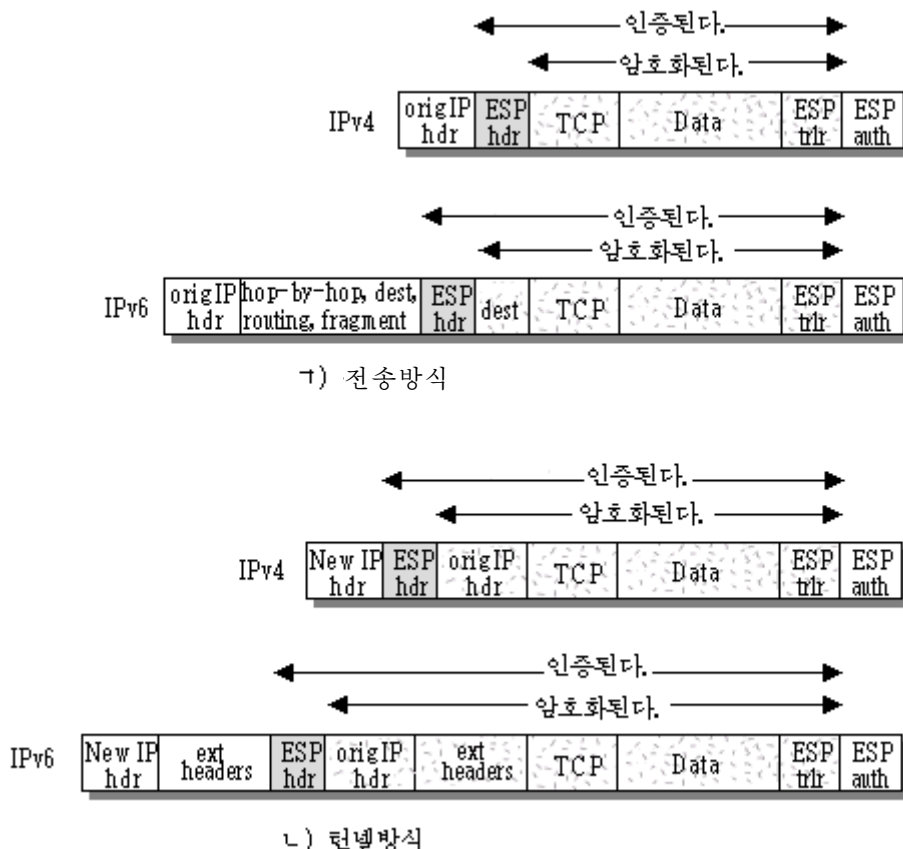



그림 13-9. ESP암호화와 인증의 범위

이제 외부의 가입자가 방화벽에 의해 보호되는 내부망우의 가입자와 통신하려고 하는 경우와 외부가입자와 방화벽들에서 ESP가 실현되는 경우를 보자.

1. 발송국은 목적하는 내부가입자의 목적지주소를 가지고 내부IP파케트를 준비한다. 이 파케트가 ESP머리부의 서두에 놓이며 다음 그 파케트와 ESP꼬리부는 암호화되고 인증자료가 부가될수 있다. 결과의 블로크는 목적지주소가 방화벽으로 되는 새로운 IP머리부(IPv6의 경우는 기본머리부에 경로정보머리부나 중계접선택 등의 확장머리부가 포함된다.)로 교갑화된다. 즉 이것은 외부IP파케트를 생성한다.
2. 외부파케트는 목적지방화벽에로 경로선택된다. 매 중개자경로조종기들은 외부IP머리부에 임의의 IP확장머리부들을 포함하여 조사하고 처리하여야 하지만 암호문을 조사할 필요는 없다.
3. 목적지의 방화벽은 외부IP머리부에 임의의 외부IP확장머리부들을 합하여 조사하고 처리한다. 다음 ESP머리부의 SPI에 기초하여 목적지마디는 그 파케트의 나머지를 복호하며 내부IP파케트의 평문을 회복한다. 다음 이 파케트는 내부망에 전송된다.
4. 그 내부파케트는 내부망의 령 또는 그이상의 경로조종기들을 통하여 경로조종되여 목적지가입자에게 전송된다.

13.5 보안련관성들의 결합

개별적SA는 AH나 ESP규약을 실현할수 있지만 둘 다 실현하지는 못한다. 때때로 특정의 전송흐름은 AH와 ESP들에 의해 제공되는 봉사들을 요구한다. 또한 특정의 전송흐름은 가입자들사이에 IPSec봉사들을 요구할수 있으며 방화벽과 같은 보안관문들사이의 개별적봉사들을 요구할수 있다.

모든 경우에 여러개의 SA들은 같은 전송흐름에 리용되여 요구하는 IPSec봉사들을 달성한다. 보안련관묶음(security association bundle)은 요구하는 IPSec봉사들의 모임을 제공하기 위해 그 전송을 처리하여야 할 SA들의 렬에 귀착된다. 그 묶음(bundle)에서 SA들은 서로 다른 끝점들이나 같은 끝점들에서 해방될수 있다.

보안련관들은 다음과 같은 두가지 방법에 의해 묶음으로 결합될수 있다.

- **전송린접성:** 같은 IP파케트에 대하여 턴넬설정에 의존하지 않고 한개 이상의 보안규약을 적용하는데 귀착시켰다. AH나 ESP를 결합하는 이 방식은 오직 한개 수준의 결합만을 허락한다. 또한 그 처리가 하나의 IPSec권고로써 수행되므로 삽입(nesting)은 아무러한 리익도 얻지 못한다.
- **반복턴넬설정:** IP턴넬설정을 통하여 여러층으로 되는 보안규약들을 적용한다는것을 지적한다. 이 방식은 개개의 턴넬이 경로상의 서로 다른 IPSec 싸이트들을 원천지로 하거나 목적지로 할수 있으므로 여러층의 삽입이 가능하다.

이 두개의 방식들은 결합될수 있다(실례로 보안관문들사이의 턴넬SA를 통하는 방법으로 가입자들사이에 전송SA를 가짐으로써).

SA묶음들을 고찰할 때 생기는 한가지 흥미 있는 문제는 인증과 암호화가 주어 진 쌍의 말단들사이에 적용될수 있다는것이다. 그에 대해 아래에 소개한다. 다음 적어도 한개의 턴넬을 포함하는 SA들의 결합을 본다.

기밀성과 인증

암호와 인증을 결합하여 가입자들사이에 기밀성과 인증을 둘 다 가지는 IP파케트를 전송할수 있다. 몇가지 방식들을 보자.

인증선택을 가지는 ESP

이 방식은 그림 13-9을 통해 잘 알수 있다. 이 방식에서 사용자는 먼저 보호할 자료에 ESP를 적용하고 다음 인증자료마당을 적용한다. 실지로 두가지 경우들이 있을수 있다.

- **전송방식의 ESP:** 인증과 암호화는 가입자에 배달되는 IP통신부하에 적용된다. 그러나 IP머리부는 보호되지 않는다.
- **턴넬방식의 ESP:** 인증은 외부IP목적지주소(실례로 방화벽)에 배달되는 IP파케트전체에 적용되며 그 목적지에서 실현된다. 내부IP파케트전체는 비밀성꾸밈새에 의하여 보호되여 내부IP목적지에로 배포된다.

다음의 두가지 경우들에 인증은 평문보다 암호문에 적용하는 편이 낫다.

전송린접성

암호화다음에 인증을 적용하는 다른 방법은 내부에 존재하는 ESP SA와 외부에 존재하는 AH SA를 가지는 두개의 SA들을 묶어서 리용하는것이다. 이 경우에 ESP는 인

증선택없이 리용된다. 내부SA는 전송SA이므로 암호화는 IP통신부하에 대하여 진행된다. 결과 파케트는 ESP에 련이은 IP머리부로 구성된다. 다음 AH는 인증이 변경되는 마당들을 내놓고 ESP에 초기의 IP머리부(및 확장들)들도 포함하도록 전송방식에 적용된다. ESP인증선택을 가지는 단일ESP SA를 간단히 리용하는데서 이 방식의 우점은 인증의 발송지와 목적지IP주소들을 포함하는 더 많은 마당들을 포함하는것이다. 결함은 한개의 SA 대 두개의 SA의 간접비용이다.

전송-턴넬묶음

암호화에 인증을 선행시키는것은 다음과 같은 몇가지 리유로 하여 적합하다. 첫째로, 인증자료가 암호화에 의해 보호되므로 누구도 그 통보문을 가로 채여 인증자료를 알지 못하며 변경시키는것도 불가능하다. 둘째로, 앞으로의 참조를 위하여 목적지에서 그 통보문을 인증정보와 함께 보관할수 있다. 만일 인증정보가 암호화되지 않은 통보문에 적용하면 이렇게 하는것은 더욱 편리하다. 다른 한편 그 통보문은 재암호화되여 인증정보를 검증해야 한다. 두 가입자들사이에 암호화전에 인증을 적용하는 한가지 방법은 내부 AH전송SA와 외부ESP련결SA로 이루어 진 묶음을 리용하는것이다. 이 경우에 인증은 변하는 마당들을 제외하고 IP머리부(및 확장)에 IP통신부하을 합하여 적용된다. 결과의 IP파케트는 ESP에 의한 련결방식으로 처리된다. 즉 그 결과는 인증된 내부파케트전체가 인증되고 암호화된 다음 새로운 외부IP머리부(및 확장)가 부가된다.

보안련관들의 기본결합

IPSec방식의 문서는 IPSec에 준하고 있는 가입자들(실례로 워크스테이션, 봉사기)이나 보안관문들(실례로 방화벽, 경로조종기)을 지원해야 할 4가지 SA들의 결합들을 서술한다. 이것들을 그림 13-10에서 보여 주었다. 매 그림의 아래부분은 요소들의 물리적련결을 표시한다. 웃부분은 하나 또는 그이상의 삽입된(nested) SA들을 경유한 론리적련결을 표시한다. 매 SA들은 AH나 ESP일수 있다. 가입자 대 가입자 SA들에 대하여 그 방식은 전송방식과 턴넬방식일수 있으며 그렇지 않으면 턴넬방식이여야 한다.

경우 1에서 모든 보안들이 IPSec를 실현하는 말단체계들사이에 제공된다. SA를 경유하여 통신하는 임의의 두 말단체계들에 대하여 그것들은 적당한 비밀열쇠를 공유하여야 한다. 가능한 결합들은 다음의것을 포함한다.

이미 이 여러가지 결합들을 인증, 암호화, 암호화전의 인증 및 암호한 후의 인증을 지원하는데 어떻게 리용할수 있는가를 론의하였다.

경우 2에 대하여 보안은 관문들사이에서만 제공되고 가입자들은 IPSec를 실행하지 않는다. 이 경우는 간단한 가상개인망지원의 례증으로 된다. 보안방식의 문서는 단일련결SA만이 이 경우에 필요된다는것을 명기한다. 턴넬은 AH, ESP 또는 인증선택권을 가지는 ESP를 지원할수 있다. 삽입된 턴넬은 IPSec봉사들이 전체 내부파케트에 적용되므로 요구되지 않는다.

ㄱ) 전송방식에서의 AH
ㄴ) 전송방식에서의 ESP
ㄷ) 전송방식(ESP SA내부나 AH SA)에서 ESP에 잇닿은 AH
ㄹ) 턴넬방식에서 AH 또는 ESP내부의 ㄱ, ㄴ 또는 ㄷ중의 임의의 하나

경우 3은 경우 2에 말단 대 말단보안을 부가하여 이룬다. 경우 1과 경우 2에서 론의된것과 같은 결합들이 여기서 허락된다. 관문 대 관문턴넬은 말단체계들사이의 모든 전

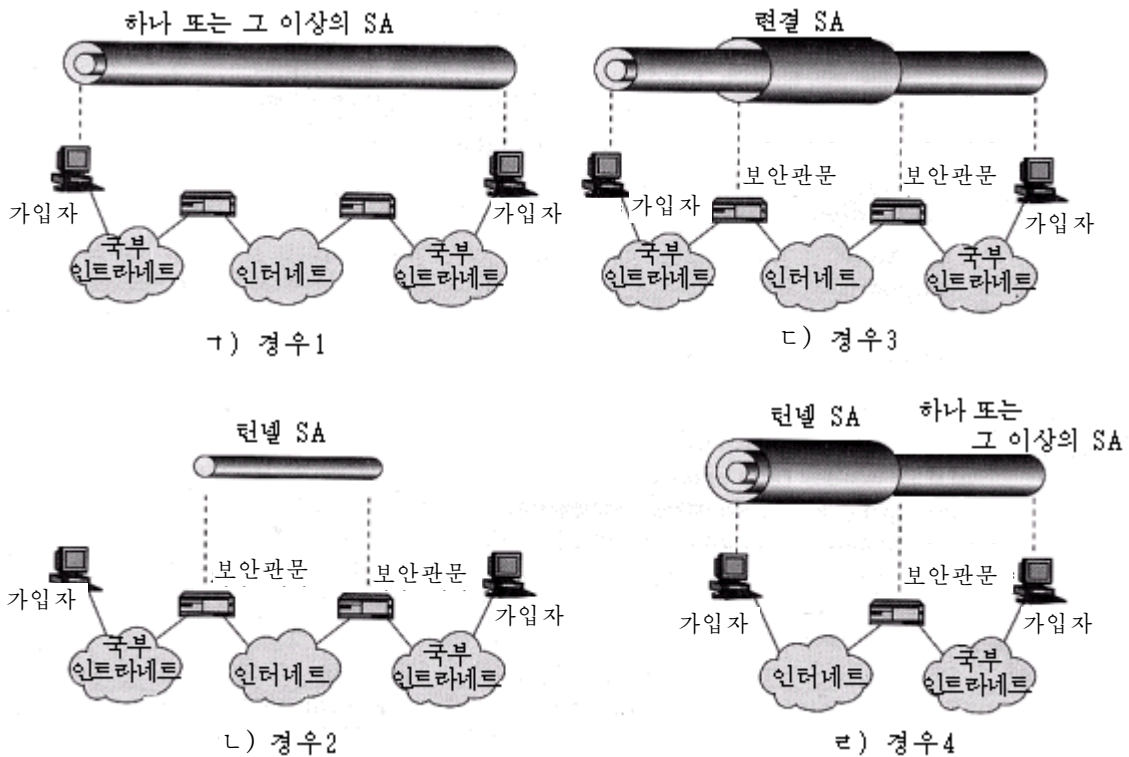



그림 13-10. 보안련관들의 기본결합

송에 대하여 인증이나 기밀성 또는 둘 다 제공한다. 관문 대 관문턴넬이 ESP일 때 이것은 전송기밀성의 제한된 형식을 제공한다. 개별적가입자들은 말단 대 말단의 SA에 의하여 응용들이나 사용자들이 요구하는 임의의 부가적인 IPSec봉사들을 실현할수 있다.

경우 4는 인터네트를 리용하여 어떤 기관의 방화벽에 도달한 다음 그 방화벽뒤의 어떤 봉사기나 워크스테이션에 접근하려고 하는 원격의 가입자를 지원하는 경우이다. 오직 턴넬방식만이 원격가입자와 방화벽사이에 요구된다. 경우 1에서처럼 하나 또는 두개의 SA들이 원격가입자와 국부가입자사이에 리용될수 있다.

13.6 열쇠관리

IPSec의 열쇠관리에는 비밀열쇠의 결정과 배포가 포함된다. 일반적인 요구조건은 두가지의 응용들 즉 AH와 ESP 둘 다에 대하여 전송과 수신쌍들사이의 통신을 위한 4개의 열쇠들이다. IPSec방식의 문서는 두가지 형식의 열쇠관리에 대한 지원을 요구한다.

- 수동: 체계관리자가 수동적으로 매 체계에 자기자체의 열쇠들과 자기와 통신하는 다른 체계들의 열쇠들을 갖추도록 한다. 이것은 비교적 작은 환경에서 현실적이다.
- 자동: 자동화된 체계는 SA들에 대하여 지령에 의한 열쇠들의 창조를 가능하게 하며 큰 분산된 체계에서 열쇠리용을 쉽게 한다.

IPSec에서 기정의 자동열쇠관리규약은 ISAKMP/Oakley라고 부르는데 다음의 요소들로 이루어 진다.

- **어우클리열쇠결정규약 :** 어우클리(Oakley)는 디피-헬만알고리듬에 기초한 열쇠교환규약인데 보안은 제공하지 않는다. 어우클리는 또한 일반적으로 특별한 형식들을 요구하지 않는다.
- **인터네트보안련관과 열쇠관리규약(Internet Security Association and key Mangement Protocol :ISAKMP):** ISAKMP자체는 특정의 열쇠교환알고리듬을 지적하지 않는다. ISAKMP는 여러가지 열쇠교환알고리듬들을 리용할수 있게 하는 통보문형태들의 모임으로 이루어 진다. 어우클리는 ISAKMP의 초기판의 리용에 위임되는 특정의 열쇠교환알고리듬이다.

어우클리에 대한 개괄로부터 시작하고 다음에 ISAKMP를 보기로 한다.

어우클리열쇠결정규약

어우클리는 디피-헬만열쇠교환알고리듬의 개선형이다. 디피-헬만은 사용자 A와 B사이의 다음의 대화를 포함한다는것을 가정한다. 두개의 대역적파라메터들인 큰 씨수 q 및 q의 원시뿌리 α에 대한 사전합의가 있어야 한다.

A는 우연수 X_A를 자기의 비밀열쇠로서 선택하고 공개열쇠 $Y_A=\alpha^{X_A}$을 B에게 전송한다. 마찬가지로 B는 우연수 X_B를 자기의 비밀열쇠로 선택하고 A에게 공개열쇠 $Y_B=\alpha^{X_B}$을 보낸다. 그 매개 혹은 비밀대화조종열쇠를 계산할수 있다.

$$K=(Y_B)^{X_A} \bmod q = (Y_A)^{X_B} \bmod q = \alpha^{X_A X_B} \bmod q$$

디피-헬만알고리듬은 두가지의 매력적인 특성들을 가지고 있다.

- 비밀열쇠는 필요할 때면 생성한다. 비밀열쇠들을 오래동안 보관할 필요가 없다.
- 교환은 대역적파라메터들에 대한 동의와 다른 하부구조가 미리 존재할것을 요구하지 않는다.

그러나 [HUIT98]에서 지적된것처럼 디피-헬만에 대한 많은 약점들이 있다.

- 대방들의 정당성에 대한 아무런 정보도 제공되지 않는다.
- 이것은 제3의 대상 C가 A와의 통신에서 B의 역을 하고 B와의 통신에서 A의 역을 하는 끼여듬공격을 받기 쉽다. A와 B는 둘 다 C와 열쇠교섭을 끝내는데 이것은 전송을 알수 없게 하고 넘겨 준다. 끼여듬공격은 다음과 같이 진행된다.

  1. B는 A의 주소로 된 통보문에 자기의 공개열쇠 X_B를 보낸다(그림 6-16을 볼것).
  2. 적(E)은 이 통보문을 가로 챈다. E는 B의 공개열쇠를 가지고 자기의 공개열쇠가 아니라 B의 사용자식별자를 가지는 통보문을 A에 보낸다. 이 통보문은 그것이 B의 가입자체계에서 온것처럼 하여 보내진다. A는 E의 통보문을 받고 E의 공개열쇠를 B의 사용자ID로 보관한다. 마찬가지로 E는 B에 대하여

A에게서 온것이라고 하는 E의 공개열쇠 붙은 통보문을 보낸다.

3. B는 자기의 비밀열쇠와 Y_E에 기초하여 비밀열쇠 K_1을 계산한다. A는 자기의 비밀열쇠와 Y_E에 기초하여 비밀열쇠 K_2을 계산한다. E는 자기의 비밀열쇠 X_E와 Y_B를 리용하여 K_1을 계산하고 X_E와 Y_A을 리용하여 K_2을 계산한다.
4. 이로부터 E는 A도, B도 자기들이 E와 통신했다는것을 모르게 A에서 B에로 또는 B에서 A에로의 통보문들을 고쳐 놓을수 있다.

- 이 알고리듬은 매우 많은 계산량을 요구한다. 결과로서 그것은 많은 열쇠들을 요구하는 혼잡(clogging)공격에 약하다. 피해자는 실지 작업보다 쓸모 없는 모드제곱을 하여 많은 계산자원들을 소비한다.

어우클리는 디피-헬만의 약점들을 제거하고 그것의 우점들을 살리도록 설계되였다.

어우클리의 특성

어우클리알고리듬은 다섯가지 중요한 성질들에 의해 특징 지어 진다.

1. 혼잡공격을 좌절시키는 쿠키로서 알려 진 구조를 리용한다.
2. 두 대방들이 어떤 그룹을 형성하는것을 가능하게 한다. 이것은 본질적으로 디피-헬만열쇠교환의 대역적파라메터들을 정의한다는것을 의미한다.
3. 재연공격을 막는데 한번쓰기정보들을 리용한다.
4. 디피-헬만의 공개열쇠값들의 교환을 가능하게 한다.
5. 중간대조공격을 막기 위하여 디피-헬만교환을 인증한다.

이미 디피-헬만을 론의하였다. 이제 이 요소들의 나머지를 차례로 보자.

먼저 혼잡공격에 대한 문제를 고찰하자. 이 공격에서 적은 정당한 사용자의 발송지주소를 위조하여 공개디피-헬만열쇠를 피해자에게 보낸다. 그러면 그 피해자는 모드제곱을 하여 비밀열쇠를 계산한다. 이 형태의 통보문들을 반복하면 쓸모 없는 작업으로 피해자의 체계를 방해할수 있다. 쿠키교환은 매개 측들이 초기통보문에서 다른 측이 인정하는 우연수인 쿠키를 보낼것을 요구한다. 그 승인은 디피-헬만열쇠교환의 첫 통보문에서 반복되여야 한다. 만일 발송지주소가 위조되였다면 적은 아무 대답도 얻지 못한다. 이리하여 적이 사용자에게 접수통지만 생성하고 디피-헬만계산을 하지 못하게 할수 있다.

ISAKMP는 쿠키생성이 다음 세개의 기본요구조건들을 만족할것을 요구한다.

1. 쿠키는 개별적대상들에 의존해야 한다. 이것은 실지의 IP주소와 UDP포구를 리용하여 쿠키를 얻고 다음 그것을 리용하여 피해자를 임의로 선택한 IP주소들 또는 포구들로 궁지에 빠뜨리는 공격을 막는다.
2. 발행자가 아닌 임의의 사람이 그 발행자가 접수할 쿠키들을 생성할수 없어야 한다. 이것은 발행자가 생성에서 국부적비밀정보와 쿠키의 이후의 검증을 리용한다는것을 의미한다. 이 비밀정보는 임의의 특정한 쿠키로부터 추론할수 없어야 한다. 이 조건들의 요점은 발행자가 그것의 쿠키들에 대한 복사를 보관할 필요가 없는데(복사를 보관하면 쿠키는 비밀발견에 보다 약해 진다.) 필요하면 들어 온 쿠키승인을 검증할수 있다.

3. 쿠키생성 및 검증방법들은 처리기자원들을 방해하려는 공격들을 막는데 민첩해야 한다.

쿠키를 창조하는 방법은 원천지와 목적지의 IP주소, UDP와 목적지포구들 및 국부적으로 생성된 비밀값들에 대하여 고속하쉬(실례로 MD5)를 진행하는것이다. 어우클리는 디피-헬만열쇠교환을 위해 서로 다른 군들을 리용한다. 매개 군들은 두개의 대역적파라메터들의 정의와 알고리듬식별자를 포함한다. 현재의 명세서에는 다음의 군들이 포함되여 있다.

- 768bit의 제곱승모드계산

$$q = 2^{768} - 2^{704} - 1 \times 2^{64} \times (\lfloor 2^{638} \times \pi \rfloor + 149686)$$
$$\alpha = 2$$

- 1024bit의 제곱승모드계산

$$q = 2^{1024} - 2^{960} - 1 \times 2^{64} \times (\lfloor 2^{894} \times \pi \rfloor + 129093)$$
$$\alpha = 2$$

- 1024bit의 제곱승모드계산
  - ¤ 파라메터들이 결정된다.
- 2^{155}이상의 타원곡선군
  - ¤ 생성기(16진): X=7B, Y=1C8
  - ¤ 타원곡선파라메터(16진): A=0, Y=7338F
- 2^{185}이상의 타원곡선
  - ¤ 생성기(16진): X=18, Y=D
  - ¤ 타원곡선파라메터(16진): A=0, Y=1EE9

첫 세개의 군들은 고전Diffie-Hellman알고리듬이 리용하는 모드에 관한 제곱이다. 마지막 두개의 군들은 6장에서 서술한 디피-헬만에 류사한 타원곡선을 리용한다.

어우클리는 재연공격에 대처하여 한번쓰기정보들을 리용한다. 매개 한번쓰기정보들은 국부적으로 생성된 모조란수이다. 한번쓰기정보들은 교환의 어떤 부분동안에 응답으로 나타나 암호화된다.

어우클리는 서로 다른 세가지 인증방법들을 리용한다.

- 수자서명: 열쇠교환은 서로 입수가능한 하쉬들을 서명하는것으로 인증한다. 즉 매개 대상들은 그 하쉬를 자기들의 비밀열쇠로 암호화한다. 그 하쉬는 사용자 ID들이나 한번쓰기정보들과 같은 중요한 파라메터들에 대하여 생성된다.
- **공개열쇠암호**: 열쇠교환은 식별자들이나 한번쓰기정보들과 같은 파라메터들을 송신자의 비밀열쇠로 암호화하여 인증된다.
- **대칭암호**: 어떤 대역밖의 기구에 의해 배포된 열쇠를 대칭암호로써 암호화하는 것으로 열쇠교환을 인증한다.

어우클리에 의한 열쇠교환실례

어우클리명세서에는 열쇠교환규약에서 리용할수 있는 많은 교환실례들이 포함되여 있다. 어우클리의 특징을 주기 위해 공격적인 열쇠교환(aggressive key exchange)이라고 부르는 하나의 실례를 보자. 세개의 통보문들만 교환되므로 그렇게 부른다.

그림 13-11에 도전적열쇠교환규약을 보여 주었다. 첫 단계에서 송신자(I)는 쿠키,

리용되는 군 및 이 교환을 위한 I의 공개디피-헬만열쇠를 전송한다. I는 이 실례에서 쓰이는 인증알고리듬 및 제공된 공개열쇠암호를 지적한다. 또한 이 통보문에는 I의 식별자들과 수신자 및 이 교환을 위한 I의 한번쓰기정보가 포함된다. 마지막으로 I는 자기의 비밀열쇠를 리용하여 두개의 식별자에 대한 서명, 한번쓰기정보, 디피-헬만공개열쇠 및 제공된 알고리듬을 첨부한다.

수신자 R가 그 통보문을 받으면 I의 공개서명열쇠로 그 서명을 검증한다. R는 군과 마찬가지로 I의 쿠키, 식별자 및 한번쓰기정보를 다시 되풀이하여 그 통보문을 확인한다. R도 역시 통보문에 이 교환을 위하여 쿠키, 자기의 디피-헬만공개열쇠, 선정된 알고리듬(제공된 알고리듬들속에 있어야 한다), 자기의 식별자, 한번쓰기정보를 포함한다. 마지막으로 R는 자기의 비밀열쇠를 리용하여 두개의 식별자, 두개의 한번쓰기정보들, 군, 두개의 디피-헬만공개열쇠 및 선정된 알고리듬에 서명을 진행하여 그 서명을 부가한다.

I가 두번째 통보문을 받으면 R의 공개열쇠로 그 서명을 검증한다. 통보문의 한번쓰기정보값에 의해 그것이 이전의 통보문의 재연이 아니라는것을 보증한다. I는 통보문을 R에게 돌려 보내여 자기가 R의 공개열쇠를 받았다는것을 검증하는것으로 이 교환을 끝맺는다.

ISAKMP

ISAKMP는 보안련관을 확립하여 교섭을 진행하고 변경 및 삭제 하기 위한 절차(procedure)들과 파케트형식들을 정의한다. SA의 설정부분으로서 ISAKMP는 열쇠교환과 인증자료교환을 위하여 통신부하을 정의한다. 이 통신부하형식들은 특정의 열쇠교환규약, 암호알고리듬 및 인증기구에 의존하지 않고 일관적으로 꾸밈새를 제공한다.

ISAKMP머리부형식

ISAKMP통보문은 하나 또는 그이상의 통신부하들이 잇닿은 ISAKMP머리부로 이루어 진다. 그것들은 모두 전송층규약에 의하여 운반된다. 명세서는 실행을 위하여 전송규약에서 UDP를 리용하여야 한다는것을 지적한다.

그림 13-12의 ㄱ에 어떤 ISAKMP통보문의 머리부형식을 보여 주었다. 그것은 다음의 마당들로 이루어 진다.

- 송신자쿠키(64bit): SA의 확립, SA통보 또는 SA삭제를 시작한 사람의 쿠키
- 수신자쿠키(64bit): 수신자의 쿠키 즉 송신자로부터의 첫 통보문에서는 무효이다.
- 다음통신부하(8bit): 통보문의 첫 통신부하의 형을 가리킨다. 통신부하에 대해서는 다음의 소절에서 론의한다.
- 기본판본(4bit): 사용중의 ISAKMP의 기본판본을 가리킨다.
- 부판본: 사용중의 부차적판본을 가리킨다.
- 교환형(8bit): 교환의 형을 가리킨다. 그에 대해 이 절의 마지막에 론의한다.기발(8bit): 이 ISAKMP교환을 위한 특정한 선택들의 설정을 가리킨다. 지금까지는 두개의 bit들이 정의되였다. 즉 암호화비트는 머리부에 따르는 모든 통신부하들이 이 SA를 위한 암호알고리듬으로 암호화되면 설정된다. 사전자료거부비트는 SA의 확립전에는 암호화된 자료를 접수하지 않도록 한다는것을 지정한다.
- 통보문ID(32bit): 이 통보문에 대한 유일한 식별자
- 통보문길이(32bit): 전체 통보문(머리부와 모든 통신부하들)의 길이

$I \rightarrow R$: $CKY, OK_KEYX, GRP, g^x, EHAO, NIDP, ID_I, ID_R, N_I, S_{KI}[ID_I || ID_R || N_I || GRP || g^x || ENAO]$

$R \rightarrow I$: $CKY_R, CKY_I, OK_KEYX, GRP, g^y, EHAS, NIDP, ID_R, ID_I, N_R, N_I, S_{KR}, [ID_R || ID_I || N_R || N_I || GRP || g^y || g^x || EHAS]$

$I \rightarrow R$: $CKY_I, CKY_R, OK_KEYX, GRP, g^y, EHAS, NIDP, ID_I, ID_R, N_I, N_R, SKI, [ID_I || ID_R || N_I || N_R || GRP || g^x || g^y || EHAS]$

기호 :

I=송신자

R=응답자

CKY_I, CKY_R=송신자, 응답자쿠키

OK_KEYX= 열쇠교환통보문형

GRP= 이 교환을 위한 Diffie-Hellman 군의 이름

g^x, g^y=송신자, 응답자의 공개열쇠: g^{xy}= 이 교환에서 대화열쇠

EHAO, EHAS= 제공 및 선택된 암호, 하쉬, 인증함수들

NIDP=암호가 이 통보문의 나머지에 쓰이지 않았다는것을 가리킨다.

ID_I, ID_R=송신자, 응답자에 대한 식별자.

N_I, N_R=이 교환에 대한 송신자, 응답자에 의해 제공된 우연

$S_{KI}[x]$, $S_{KR}[x]$=송신자, 응답자의 비밀열쇠(서명열쇠)를 리용하는 X 에 대한 서명임을 가리킨다.

그림 13-11. Aggressive Oakley Key Exchange 의 실례

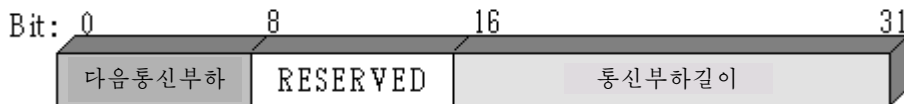
ㄱ) ISAKMP머리부

ㄴ) 일반통신부하머리부

그림 13-12. ISAKMP형식

ISAKMP통신부하형태

모든 ISAKMP통신부하들은 그림 13-12의ㄴ에서 볼수 있는것처럼 같은 일반머리부로 시작된다. 다음통신부하마당은 이것이 그 통보문의 마지막통신부하이면 0값을 가지고 아니면 그 값은 다음의 통신부하의 형을 가리킨다.

표 13-3에 ISAKMP에 대하여 정의된 통신부하형들을 개괄하고 매 통신부하들의 부분인 마당들 또는 파라메터들을 렬거하였다. SA통신부하은 SA의 설정을 시작하는데 쓰인다. 이 통신부하에서 해석파라메터의 령역은 협상이 진행중에 있는 DOI를 식별한다. IPSec DOI가 그 하나의 실례인데 ISAKMP는 다른 문맥으로도 쓸수 있다. 상태(Situation)파라메터는 이 협상에 대한 보안방략을 정의한다. 즉 본질적으로 암호화와 기밀성에 요구되는 보안수준들이 렬거된다.

제안통신부하(Proposal Payload)은 SA발행기간에 쓰이는 정보를 포함한다. 통신부하은 봉사들과 기구들을 협상하고 있는 SA의 규약(ESP 또는 AH)을 가리킨다. 통신부하은 또한 송신자의 SPI나 변환의 수도 포함한다. 매개 변환들은 변환통신부하에 포함된다. 다중변환통신부하들의 리용은 송신자가 몇가지 가능성(능력)을 제공할수 있게 해주는데 수신자는 그중 하나를 선택하거나 제공을 부결할수 있다.

변환통신부하(transform payload)은 지정된 규약에 의해 송신통로를 안전하게 하는데 쓰이도록 보안변환을 정의한다. 변환번호파라메터는 수신자가 특정의 통신부하을 리용하여 그 변환의 접수를 지적할수 있도록 통신부하을 확인하는데 리용된다. 변환- ID와 속성마당들은 구체적인 변환(ESP에는 3중DES, AH에는 HMAC-SHA-1-96)을 그것과 관련한 속성들(실례로 하쉬길이)을 가지고 확인한다.

표 13-3. ISAKMP Payload 형들

형	파라메터들	해 설
Security Associaition	Domain of Interpretation, Situation	보안속성들을 참의하고 그 합의가 이루어 진 기초에서 DC Situation을 가리키는데 리용된다.
Proposal(P)	Proposal#, Protocol-ID, SPI Size, #of Transforms, SPI	SA합의동안 리용된다. 리용되는 규약은 변환들의 수를 지적한다.
Transform(T)	Transform#, Transform-ID, SA Attributes	SA합의동안 리용된다. 변환과 관련되는 SA속성들을 지적한다.
Key Exchange(KE)	Key Exchange Data	여러가지 열쇠교환기술들을 지원한다.
Identification(ID)	ID Type, ID Data	확인정보를 교환하는데 리용된다.
Certificate(CERT)	Cert Encoding, Certificate Data	증명서들과 다른 증명성관련정보를 전송하는데 리용된다.
Certificate Request(CR)	#Cert Types, Certificate Types, #Cert Auths, Certificate Authorities	증명서를 요구하는데 쓰인다. 요구하는 증명서의 형들과 접수가능한 증명국들을 지원한다
Hash(HASH)	Hash Data	하쉬함수에 의해 생성되는 자료를 포함한다.
Signature(SIG)	Signature Data	서명함수에 의해 생성되는 자료를 포함한다.
Nonce(NONCE)	Nonce Data	Nonce를 포함한다.
Notification(N)	DOI, Protocol-ID, SPI Size, Notify Message Type, SPI, NotificationData	오유조건과 같은 통보자료를 전송하는데 쓰인다.
Delete(D)	DOI, Protocol-ID, SPI Size, #of SPIs, SPI(one or more)	더는 쓸수 없는 SA를 지적한다.

열쇠교환통신부하(Key Exchange Payload)은 어우클리, 디피-헬만 및 PGP에 쓰이는 RSA기초의 열쇠교환들을 비롯한 여러 열쇠교환기술들에 쓸수 있다. 열쇠교환자료마당은 대화열쇠를 생성하는데 요구되는 자료를 포함하며 열쇠교환알고리듬에 의하여 결정된다.

식별통신부하(Identification Payload)은 통신하는 대방의 신원을 결정하는데 리용되며 정보의 확실성을 결정하는데 리용할수 있다. 일반적으로 ID Data마당은 IP∨4 또는 IP∨6주소를 포함한다.

증명서통신부하(Certificate Payload)은 공개열쇠의 증명서를 나른다. 그 Certificate Encoding마당은 증명서의 형이나 증명서관련의 정보를 가리킨다. 거기에는 다음의 것들이 포함될수 있다.

- PKCS #7 을 리용한 X.509증명서
- PGP증명서
- 서명된 DNS열쇠
- X.509증명서-서명
- X.509증명서-열쇠교환
- Kerberos의 표식
- 증명서취소목록(Certificate Revocation List:CRL)
- 권한취소목록(Authority Revocation List:ARL)
- SPKI증명서

ISAKMP교환의 임의의 순간에 송신자는 다른 통신상대의 증명서를 요구하는 **증명서요구통신부하(Certificate Request Payload)**을 포함할수 있다. 그 통신부하은 접수가능한 한개이상의 증명서형과 접수가능한 한개이상의 증명국을 렬거할수 있다.

하쉬통신부하(Hash Payload)에는 통보문의 일부와 ISAKMP상태를 하쉬한것 또는 ISAKMP의 상태만을 하쉬함수로 생성한 자료가 들어 간다. 이 통신부하은 통보문중의 자료의 완정성을 검증하는데 리용할수 있는데 부인불가능한 봉사를 제공하는데도 리용할수 있는 가능성이 있다.

서명통신부하은 통보문 및/또는 ISAKMP의 일부 상태들에서 수자서명함수에 의하여 생성된 자료를 포함한다. 이 통신부하은 통보문자료의 완정성을 검증하는데는 리용되지만 비거절봉사들에는 리용할수 없다.

한번쓰기정보통신부하(Nonce Payload)은 교환기간에 그 유효성을 담보하며 재연공격을 막는데 리용되는 우연자료를 포함한다.

통보통신부하(Notification Payload)에는 해당 SA 또는 해당 SA협상과 관련한 오유나 상태정보가 포함된다.

다음의 ISAKMP오유통보문들이 정의되여 있다.

Invalid Payload Type	Invalid Protocol ID	Invalid Certificate
DOI Not Supported	Invalid SPI	Bad Cert Request Syntax
Situation Not Supported	Invalid Transform ID	Invalid Cert Authority
Invalid Cookie	Attributes Not Supported	Invalid Hash Information
Invalid Major Version	Not Proposol chosen	Authentication Failed
Invalid Minor Version	Baol Proposal Syntax	Invalid Signature
Invalid Exchange Type	Payload Malformed	Address Notification
Invalid Flags	Invalid Key Information	
Invalid Message ID	Invalid Cert Encoding	

지금까지 정의된 ISAKMP상태통보문만이 접속된다. 이 ISAKMP통보들외에 DOI-특유의 통보들이 쓰인다. IPSec에 대하여 다음의 부가적인 상태통보문들이 정의된다.

- **수신자의 생명주기**(Responder-Lifetime): 수신자가 선택한 SA의 생명주기를 알려 준다.
- **재연상태**(Replay-Status): 수신자가 반재연적발을 하겠는가 않겠는가 하는 수신자의 선택을 명백히 확인하는 경우에 쓰인다.
- **첫 접촉**(Initial-Contact): 대방에게 이것이 그 체계에서 확립되는 첫 SA라는 것을 알려 준다. 이 통보의 접수자는 송신측의 체계가 재기동되며 역시 기존의 SA에 접근하지 않는다고 가정하며 그때 송신측체계와의 모든 SA들을 해소할 수 있다.

소거통신부하(Delete Payload)은 송신자가 그 자료기지로부터 제거하며 따라서 더 이상 유효하지 않는 하나 또는 그이상의 SA들을 가리킨다.

ISAKMP통보문교환

ISAKMP는 통보문교환을 위한 구조에 기초블로크로서 봉사하는 통신부하형들을 제공한다. 그 명세서는 다섯개의 기정의 교환형들을 지원한다. 즉 이것들을 표 13-4에서 개괄하였다. 표에서 SA는 관련된 규약들과 변환통신부하들을 가리킨다.

기본통보문교환(Base Exchange)에서는 열쇠교환과 인증자료가 함께 전송된다. 이것은 신원보호를 제공하지 않는 대신 교환의 회수를 최소화한다. 첫 두개의 통보문들은 쿠키들을 제공하고 SA를 동의한 규약과 변환들을 제정한다. 쌍방은 한번쓰기정보를 리용하여 재연공격에 대항한다. 마지막 두개의 통보문들에서는 열쇠관련정보들,신원정보외에 처음 두개의 통보문들로부터의 열쇠,사용자ID,한번쓰기정보들을 인증하기 위하여 쓰이는 AUTH통신부하를 교환한다.

신원보호통보문교환(Identity Protection Exchange)에서는 기본통보문교환을 확장하여 사용자들의 신원을 보호한다. 첫 두 통보문들에서 SA를 제정한다. 다음 두개의 통보문들은 열쇠교환의 재연공격에 대항하기 위해 한번쓰기정보들과 함께 열쇠를 교환한다. 대화열쇠가 일단 계산되면 쌍방은 수자서명이나 경우에 따라서는 공개열쇠들을 보증하는 증명서라고 하는 인증정보를 포함하는 암호화된 본문들을 교환한다.

통보문교환에만 기초한 인증(Authentication Only Exchange)은 열쇠교환을 하지 않고 호상인증을 하는데 쓰인다. 첫 두 통보문들은 SA를 확립한다. 그외 수신자는 두번째 통보문을 리용하여 그것의 ID를 나르고 인증을 리용하여 그 통보문을 보호한다. 송신자는 세번째 통보문을 보내여 그것의 인증된 ID를 전송한다.

공격적인 통보문교환(Aggressive Exchange)에서는 신원에 대한 보호를 제공하지 않는 대신에 교환의 회수를 최소화한다. 첫 통보문에서 송신자는 규약과 변환선택들을 제시하여 SA를 제안한다. 또한 송신자는 열쇠교환을 시작하고 자기의 ID를 보낸다. 두번째 통보문에서 수신자는 특정의 규약과 변환을 가지는 SA의 접수를 알리고 열쇠교환을 끝내며 전송된 정보를 인증한다. 세번째 통보문에서 송신자는 공유된 비밀대화열쇠로 암호화된 앞의 정보를 포함하는 인증결과를 전송한다.

정보의 교환(Informational Exchange)은 SA관리를 위한 정보의 한방향전송에 쓰인다.

표 13-4. ISAKMP 의 통보문교환형태

교환	주목
ㄱ)기본통보문교환	
(1) I→R: SA; NONCE	ISAKMP-SA협의를 시작한다.
(2) R→I: SA; NONCE	기본SA가 합의된다.
(3) I→R: KE; ID_I;AUTH	열쇠가 생성된다. 송신자의 신원이 수신자에 의해 검증된다.
(4) R→I: KE; ID_R;AUTH	수신자신원이 송신자에 의해 검증, 열쇠가 생성되고 SA가 확립된다.
ㄴ) 신원보호통보문교환	
(1) I→R: SA;	ISAKMP-SA협의를 시작한다.
(2) R→I: SA	기본SA가 합의된다.
(3) I→R: KE; NONCE	열쇠가 생성된다.
(4) R→I: KE; NONCE	열쇠가 생성된다.
(5)* I→R: KE; ID_I;AUTH	송신자의 신원이 수신자에 의해 검증된다.
(6)* R→I: ID_R;AUTH	수신자신원이 송신자에 의해 검증된다. SA가 확립된다.
ㄷ) 인증만에 의한 통보문교환	
(1) I→R: SA; NONCE	ISAKMP-SA협의를 시작한다.
(2) R→I: SA;NONCE;ID_R;AUTH	기본SA가 합의된다.수신자신원이 송신자에 의해 검증된다.
(3) I→R: ID_I;AUTH	수신자신원이 송신자에 의해 검증된다. SA가 확립된다.
ㄹ) 공격적인 통보문교환	
(1) I→R: SA;KE;NONCE;ID_I	ISAKMP-SA협의와 열쇠교환을 시작한다.
(2) R→I: SA;KE;NONCE;ID_R;AUTH	송신자의 신원이 수신자에 의해 검증된다. 열쇠가 생성된다. 기본SA가 합의된다.
(3) I→R: AUTH	수신자의 신원이 송신자에 의해 검증된다. SA가 확립된다.
ㅁ) 통보의 교환	
(1)* I→R: N/D	오유나 상태통지 또는 삭제

기호:

I= 송신자

R= 수신자

* = ISAKMP 머리부다음의 통신부하는 암호화된다

참고문헌

IPv6과 IPv4는 문헌 [STAL97]에서 더 자세히 서술되였다. 인터네트봉사와 IPv4의 성공적인 적용은 문헌 [COME95]와 [STEV94]에서 ,IPv6에 관련한 문제들은 문헌 [BRAD96]에서 취급하였다. 문헌 [HUIT98]은 모두 IPv6의 명세를 구성하는 여러가지 RFC들의 기술적서술이다. 문헌 [CHIN98]은 IPSec설계에 대한 훌륭한 론의를 제공한다.

BRAD96 Bradner, S., and Mankin, A. *Ipng: Internet Protocol Next Generation*. Reading, MA: Addison-Wesley, 1996

CHIN98 Chery. P., et al. "A Security Architecture for the Internet Protocol." *IBM Systems Journal*, Number 1, 1998

COME 95a Comer, D. *Internetworking with TCP/IP, Volume I; Principles, Protocols and Architecture*. Upper Saddle River, NJ: Prentice Hall, 1995

HUIT98 Huitema, C. *IPv6: The New Internet Protocol*. Upper Saddle River, NJ: Prentice Hall, 1998

STAL97 Stallings, W. *Data and computer Communications, Fifth Edition*. Upper Saddle River, NJ: Prentice Hall, 1997

STEV94 Steveus, W. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison_Wesley, 1994

참고할 Web싸이트들

- IP Security Drotocol (ipsec)Charter: 최신 RFC들과 IPSec에 대한 인터네트대본들
- IP Security Working Group News: 작업그룹문서들 우편기록, 관련기술론문들 및 기타 자료
- IP Security (IPSEC) Resoruces: IPSec를 실현하는 회사들에 대한 목록 및 기타 자료

문 제

1. AH처리를 론하는데서 IP머리부의 모든 마당들이 MAC계산에 리용되지 않는다는 것이 취급되였다.
   1) IPv4머리부의 매개 마당들에 대하여 그 마당이 불변 또는 가변이지만 예측가능한가 그렇지 않으면 가변인가를 설명하시오.
   2) IPv6머리부에 대하여 우의 요구에 대답하시오.
   3) IPv6확장머리부들에 대하여 우의 요구에 대답하시오.

   매 경우에 대하여 그 매 마당을 선택한 리유를 밝히시오.

2. 턴넬방식을 리용할 때 새로운 외부IP머리부가 구성된다. IPv4와 IPv6 둘 다에 대하여 외부와 내부의 각각의 외부IP머리부마당과 외부파케트의 매 확장머리부의 사이관계를 밝히시오. 즉 어느 외부값들이 내부값들에서 유도되며 내부값들에 무관계하게 구성되는가를 지적하시오.
3. 말단 대 말단인증과 암호는 두 가입자들사이에서 요구된다.
   1) 인증하기전에 암호화가 적용되는 전송린접
   2) 인증하기전에 암호화가 적용되는 턴델 SA내부에서 묶음된 전송 SA
   3) 암호화하기전에 인증이 적용되는 턴델 SA내부에서 묶음된 전송 SA
   을 보여 주는 그림 13-6과 13-9와 류사한 그림을 그리시오.
4. IPSec방식의 문서는 두개의 전송방식 SA들이 같은 말단 대 말단흐름우의 AH와 ESP의 두 규약들을 허락하도록 한다.
5. 1) ISAKMP교환형들(표 13-4)중에서 어느것이 공격적인 어우클리열쇠교환(그림 13.11)에 대응하는가?
   2) 공격적인 어우클리열쇠교환에 대하여 매 통보문에서의 어느 파라메터들이 어느 ISAKMP통신부하형들에 들어 가는가를 지적하시오.

부록 13: 호상련결망과 인터네트규약

이 부록에 인터네트규약들에 대하여 개괄하였다. 먼저 호상련결망을 제공하는데서 인터네트규약의 역할에 대하여 개괄한다. 다음 두개의 기본 호상련결망규약들인 Ipv4와 Ipv6를 소개한다.

인터네트규약의 역할

인터네트규약(IP)은 여러개의 망들을 거쳐 말단체계들을 호상접속하는 기능을 제공한다. 이를 위하여 IP가 매개 말단체계들과 경로조종기들에 장비된다. 원천지말단체계에서 더 높은 준위의 자료는 전송을 위해 IP규약자료단위(PDU)로 교갑화된다. 다음 이 PDU는 하나 또는 그이상의 망들과 련결된 경로조종기들을 통과하여 목적지말단체계에 도달한다.

경로조종기는 다음의것들을 포함하는 망들속에서 여러가지 차이들을 처리할수 있어야 한다.

- **주소화방식들:** 망은 장치들에 주소들을 할당할 때 서로 다른 방식들을 리용할수 있다. 실례로 IEEE 802 LAN접속한 매개 장치들에 16-bit 또는 48-bit2진주소들을 리용하며 X.25공개파케트-절환망은 12자리 10진주소들을 리용한다(48-bit 주소에 대하여 자리당 4bit로 부호화된다). 지역망주소화의 일부 형식이 등록부봉사와 마찬가지로 제공되여야 한다.
- **최대파케트크기들:** 한개의 망으로부터 파케트들은 더 작은 쪼각들로 쪼개져 다른 망으로 전송될수 있다. 이때의 처리를 쪼각화(fragmentation)라고 부른다. 실례로 에써네트(Ethernet)에서는 최대파케트크기가 1500byte, X.25망들에서는 최대파케트의 크기가 1000byte인것이 보통이다. 에써네트체계에서 전송되고

X.25망에서의 재전송을 위하여 경로조종기에 의해서 지정되는 파케트는 들어오는 파케트들을 두개의 보다 작은것들로 쪼갤수 있게 되여야 한다.

- **대면부들:** 여러 망들에 대한 하드웨어와 쏘프트웨어대면부들은 서로 다르다. 경로조종기의 개념은 이 차이들과 독립이여야 한다.
- **믿음성:** 여러가지 망봉사들은 믿음성 있는 말단 대 말단가상회로부터 믿을수 없는 봉사까지의 임의의것을 제공할수 있다. 경로조종기들의 조작은 망의 믿음성에 대한 가정에 의존하지 말아야 한다.

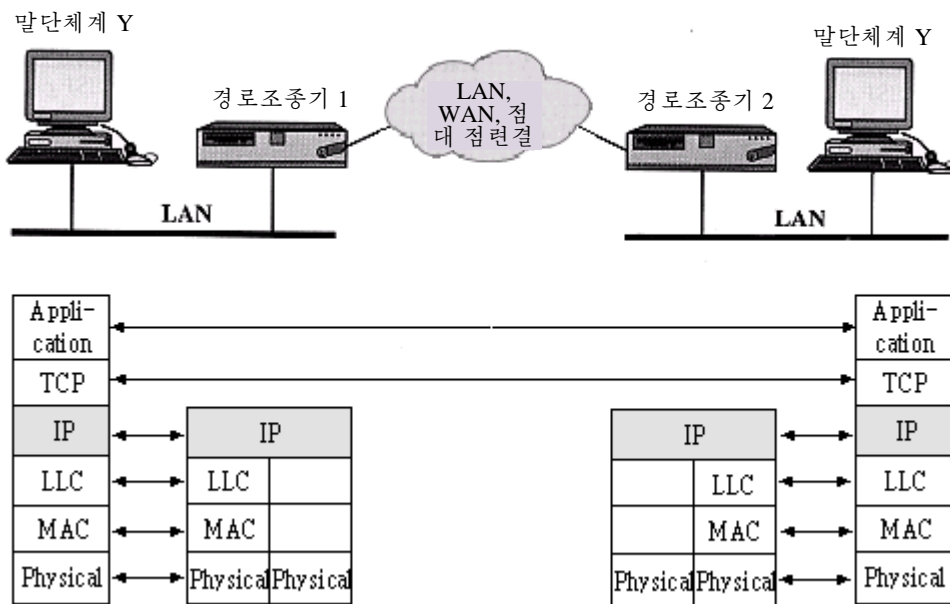



그림 13-13. TCP/IP에 대한 구성실례

경로조종기의 조작은 그림 13-13에서와 같이 인터네트규약에 의존한다. 이 실례에서는 TCP/IP규약조(suite)의 인터네트규약(IP)이 그 기능을 수행한다. IP는 경로조종기들과 마찬가지로 모든 망들의 매 말단체계들에서 실행되여야 한다. 또한 매 말단체계는 원만히 련결되기 위하여 IP상의 호환규약들을 가져야 한다. 그림 13-13에서는 말단체계 X로부터 말단체계 Y에로 자료블로크의 전송을 고찰하였다. X에서 IP층은 X의 TCP로부터 Y에 보낼 자료블로크들을 접수한다. IP층은 Y의 대역인터네트주소들을 서술하는 머리부에 붙는다. 그 주소들은 두개의 부분들 즉 망식별자나 말단체계식별자속에 있다. 이 블로크를 IP파케트로서 귀착시킨다. 다음으로 IP는 목적지(Y)가 다른 부분망에 있다고 인정한다. 따라서 첫 단계는 파케트를 경로조종기에 보내는것이다. 이 경우에는 경로조종기 1에 보낸다. 이를 위해 IP는 그 자료단위를 적당한 주소정보와 함께 LLC에 넘겨준다. 그러면 LLC는 MAC에 넘겨 줄 LLC PDU를 창조한다. MAC층은 머리부가 경로조종기 1의 주소를 가지는 MAC파케트를 창조한다. 다음 그 파케트는 LAN을 거쳐 경로조종기 1로 간다.

경로조종기는 그 파케트와 LLC머리부 및 꼬리부들을 제거하고 IP머리부를 해석하여 그 자료의 최종목적지(이 경우 Y)를 결정한다. 이때 경로조종기는 경로조정결심을 내려야 한다. 두가지 가능성들이 있다:

1. 목적지의 말단체계 Y는 경로조종기가 붙은 부분망들중의 하나에 직접 접속된다.
2. 목적지에 도달하기 위하여 하나 또는 그이상의 부가되는 경로조종기들을 통과하여야 한다.

이 실례에서 파케트는 목적지에 도달하기전에 경로조종기 2를 통하여 경로조종되여야 한다. 따라서 경로조종기 1은 중개망을 경유하여 경로조종기 2에 IP파케트를 보낸다. 이때 그 망에서의 규약들이 리용된다. 실례로 중개망이 X.25망이면 IP자료단위는 적당한 주소정보와 함께 X.25파케트로 포장되여 경로조종기 2에 도달한다. 이 파케트가 경로조종기 2에 도달하면 그 파케트머리부는 해제된다. 경로조종기는 이 IP파케트가 경로조종기가 붙어 있는 부분망에 직접 접속된 Y에 보내진다는것을 결정한다. 따라서 경로조종기는 Y의 목적지주소를 가지고 파케트를 생성하여 LAN으로 내보낸다. 마지막에 그 자료는 Y에 도착하여 거기에서 파케트, LLC 및 인터네트머리부들과 꼬리부들이 해제된다.

IP에 의해 제공되는 이 봉사는 믿을수 없는것이다. 즉 IP는 모든 자료들이 배달되거나 그 배달된 자료가 도착하는것이 규칙적으로 진행된다는것을 담보하지는 않는다. 발생된 오유로부터 회복은 다음의 웃층의 책임이다(이 경우 TCP). 이 수법은 큰 유연성을 제공한다. 배달이 담보되지 않으므로 임의의 부분망들에서 특정한 믿음성요구는 없다. 따라서 규약은 부분망형들의 어떤 조합으로 가동할것이다.

배달렬(the sequence of delivery)이 담보되지 않으므로 다음의 파케트들은 서로 다른 경로들을 따라 인터네트를 통과할수 있다. 이것은 규약이 경로들을 변경시켜 호상망에서의 혼란과 실패에 대처할수 있게 한다.

IPv4

수십년동안 IP판본 4는 TCP/IP규약구성방식의 초석으로 되여 왔다. 그림 13-14에 최소 20octet 또는 160bit의 IP머리부를 보여 주었다. 그 마당들은 다음과 같다.

Version(4bit): 규약의 발행을 제시하는 판번호를 지정한다. 그 값은 4이다.

Internet Header Length(THL)(4bit): 머리부의 길이는 32-bit단어이다 . 최소값은 20octect의 최소머리부길이일 때 5이다.

Type of Service(8bit): 파케트의 상대적우선권으로 말단체계 IP모듈들과 파케트의 경로에 따르는 경로조종기들에 대한 지도서로 제공된다.

Total Length(16bit): 옥테드로 표시된 총 IP파케트의 길이이다.

Identification(16bit): 발송지주소,목적지주소 및 사용자규약과 함께 파케트를 유일하게 식별하기 위한 렬번호. 따라서 식별자는 파케트의 발송지주소, 목적지주소 및 그 파케트가 인터네트에 남아 있는 시간에 대하여 유일해야 한다.

Flags(3bit):매 시점에서 오직 두 비트만이 정의된다. 파케트가 쪼각화되였을 때 More비트는 이것이 원래의 파케트에서 마지막쪼각인가 아닌가를 나타낸다. Don’t Fragment bit가 설정되면 쪼각화를 금지한다. 목적지가 그 쪼각들을 재조합할 능력을 가지고 있지 못하다는것이 알려 지는 경우 쓸모 있다. 그러나 이 비트가 설정되였을

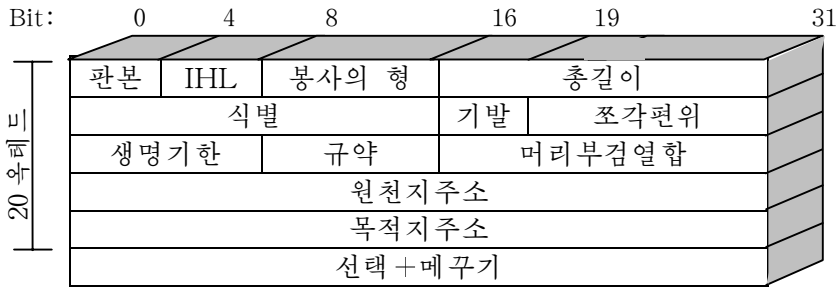
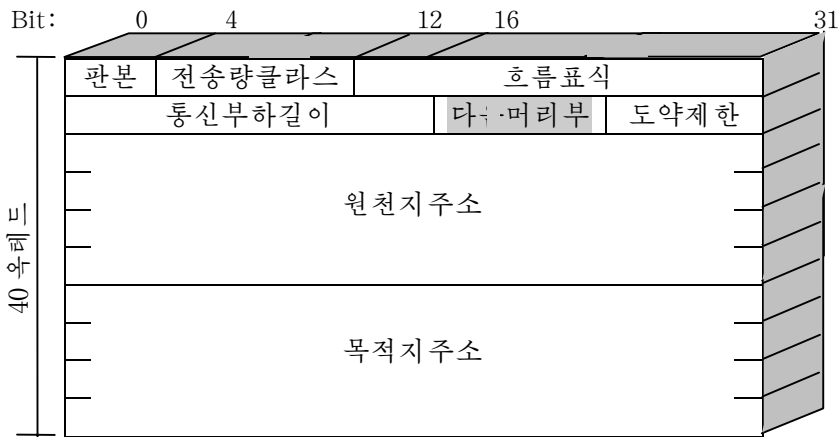
때 그 파케트가 부분망경로의 최대크기를 초과하면 그것을 버린다. 따라서 그 비트가 설정되면 최대파케트크기가 작은 부분망들을 피하기 위해 원천경로조종방법을 리용할 수 있다.

ㄱ) Ipv4머리부

ㄴ) Ipv6머리부

그림 13-14. IP머리부들

- Fragment Offset(13bit): 원래 파케트에서 이 쪼각이 속하는 위치를 64-bit단위로 측정하여 지적한다. 이것은 마지막쪼각이 아닌 쪼각들은 길이가 64bit의 배수인 자료마당을 포함해야 한다는것을 암시한다.
- Time to Live(8bit):파케트가 인터네트에 얼마나 오래 남아 있게 되는가를 초의 시간단위로 지적한다. 파케트를 처리하는 모든 경로조종기들은 TTL을 최소 1까지 감소시켜야 하며 따라서 TTL은 도약계수와 어느 정도 류사하다.
- Protocol(8bit): 목적지에서 자료마당을 받을 다음 웃준위규약을 지정한다. 따라서 이 마당은 IP머리부단과 그 파케트의 다음머리부의 형을 식별한다.
- Header Checksum(16bit): 머리부에만 적용되는 오유탐색코드
- 일부 머리부마당들이 전송중에 변할수 있으므로 이것은 매 경로조종기에서 재검증 및 재계산된다. 검사합마당은 머리부의 모든 16bit단어들의 합이다. 계산을 위해 검사합마당은 그자체가 령값으로 초기화된다.

- **source Address(32bit):** 망과 지정된 망에 붙은 말단체계를 서술하기 위해 비트들의 가변배정을 할수 있도록 부호화된것.
- **Destination Address(32bit):** 발송시주소와 같은 특성을 가진다.
- **Options(가변):** 송신하는 사용자가 요구하는 선택들을 부호화한다. 이것들도 보안표식,원천경로조종,레코드경로조종 및 시간표식을 포함할수 있다.
- **Padding(가변):** 그 파케트머리부의 길이가 32bit의 배수가 되도록 하는데 리용된다.

IPv6

1995년에 IETF(Internet Engineering Task Force)는 IPng로 알려 진 다음 세대의 IP에 대한 명세서로 발행된 인터네트용의 규약표준을 개발하였다. 이 명세서(Specification)는 1996년에 IPv6으로 알려 진 표준으로 되였다. IPv6은 현대의 IP(IPv4로 알려 진것)보다 많은 기능적향상들을 가져 왔으며 더욱 현실로 되고 있는 화상과 비데오를 포함하여 자료흐름의 혼합과 오늘날의 망들을의 더 고속화하도록 설계되였다. 그러나 새로운 규약의 개발을 뒤받침해 준 추동력이 바로 보다 많은 주소들에 대한 요구였다. IPv4는 발송지와 목적지를 서술하는데 32-bit주소를 리용한다. 인터네트와 인터네트에 접수한 전송망들의 폭발적인 증대로 하여 이 주소길이는 주소들을 요구하는 모든 체계들의 편리를 도모하는데 불충분하게 되였다. 그림 13-14에서와 같이 IPv6은 128-bit원천지와 목적지주소마당들을 가진다.

종국적으로 ICP/IP를 리용하는 모든 설정들은 현대의 IP를 IPv6으로 고칠것을 예견하고 있는데 이것은수십년까지는 않되지만 몇년 잘 걸릴것이다.

IPv6머리부

IPv6머리부는 40octet의 고정된 길이를 가진다. 그것은 다음의 마당들로 이루어 진다(그림 13-14의 b).

- **Version(4bits):** 인터네트규약판본번호:값은 6이다.
- **Traffic Class(8bits):** IPv6파케트들의 서로 다른 클라스들이나 우선권 등을 확인하고 식별하기 위하여 마디들을 생성하거나 경로기들에 전송함으로써 리용할수 있다. 이 마당에 대한 리용은 아직 연구중에 있다.
- **Flow Label(20bits):** 망내에서 경로조종기들에 의한 특수한 조종을 요구하는 파케트들을 표시하기 위하여 가입자가 리용할수 있다.흐름표식화(Flow labeling)에 의해 원천지예약과 실시간 전송처리가 방조될수 있다.
- **Payload Length(16bits):** 머리부에 이은 IPv6파케트의 나머지부분의 길이(옥테드단위로). 다시말하여 이것은 모든 확장머리부들과 전송준위PDU의 전체 길이이다.
- **Next Header(8bits):** IPv6머리부에 련이은 머리부의 형을 식별한다. 이것은 IPv6확장머리부이거나 TCP 또는 UDP와 같은 상위층머리부이다.
- **Hop Limit(8bits):** 이 파케트에 대해 허락되는 도약들의 나머지수. 도약제한은 원천지에서 요구하는 최대값으로 설정되며 그 파케트를 전송하는 매개 마디에 의해 하나씩 감소된다. 도약제한이 령으로 되면 그 파케트는 철회된다.
- **Source Address(128bits):** 그 파케트작성자의 주소

- Destination Address(128bits): 파케트수신자의 주소. 이것은 사실 경로조종 확장머리부가 존재하면 최종목적지로 될수 없다.

IPv6머리부가 IPv4의 mandatory portion이 보다 길지만(40octet대 20octet) 거기에 포함되는 마당의 수는 오히려 적다(8대12). 따라서 경로조종기들도 머리부당 처리량이 적어 지는데 이로써 경로조종속도가 높아 진다.

IPv6 확장머리부

IPv6파케트는 IPv6머리부와 령 또는 그 이상의 확장머리부들을 포함한다.
IPSec의 밖에서 다음의 확장머리부들이 정의되고 있다.

Hop-by-Hop Options header: hop-by-hop처리를 요구하는 특수선택들을 정의한다.
Ronting header: IPv4원천경로조종과 마찬가지로 확장된 경로조종을 제공한다.
Fragment header: 쪼각화와 재조합정보를 포함한다.
Authentication header:파케트안전성과 인증을 제공한다.
Encapsulating Security Payload header:개인성을 제공한다.
Destination Options header:목적지마디가 검사하는 선택정보를 포함한다.

IPv6표준에서는 여러개의 확장머리부들이 리용될 때 IPv6머리부들이 다음의 순서로 나타난다.

1. IPv6머리부: 항상 제일 먼저 나타나야 한다.
2. Hop-by Hop 선택머리부
3. 목적지선택머리부: IPv6 목적지주소마당에 나타나는 첫 목적지와 경로조종머리부에서 명기된 부분렬 목적지들에 의해 처리된다.
4. 경로조종머리부
5. 쪼각머리부
6. 인증머리부
7. 교갑화보안통신부하머리부
8. 목적지선택머리부: 파케트의 마지막목적지에 의해서만 처리되는 선택들에 한함

그림 13-15에 매개 비보안머리부의 구체례를 포함하는 IPv6파케트의 실례를 보여 주었다. IPv6머리부는 매개 확장머리부가 머리부마당을 포함한다는것을 주의해 둔다.
이 마당은 련이은 머리부의 형을 식별한다. 만일 다음 머리부가 확장머리부이면 이 마당은 그 머리부의 형식별자를 포함한다. 그렇지 않으면 이 마당은 IPv4 규약마당과 같은 값에 의하여 IPv6을 리용하는 상위층규약의 규약식별자를 포함한다.
그림 13-15에서의 상위층규약도 TCP이며 따라서 IPv6파케트에 의해 전송된 상위층 자료는 응용자료블로크에 잇닿은 TCP머리부로 구성된다.
도약별 선택머리부는 만일 존재하면 경로에 따라서 모든 경로조종기들에 의해 검사하여야 할 선택정보를 나른다. 그 머리부에는 다음의 마당들이 포함된다.

- Next Header(8bit): 이 머리부가 잇닿은 머리부형을 식별한다.
- Header Extension Length(8bit): 첫 64bit를 포함하지 않는 64-bit단위로 표

시된 이 머리부의 길이

- **Options:** 하나 또는 그 이상의 선택들을 포함한다. 매 선택은 세개의 부분마당 즉 선택형을 지적하는 tag, 길이 및 값으로 이루어 진다.

지금까지 한개의 선택 $2^{16}-1=65535$octet보다 더 긴 통신부하들을 가진 IPv6파케트들을 보내는데 리용되는 Jumbo통신부하만이 정의되였다. 이 선택의 Option Data 마당은 32bit이며 그 파케트의 길이를 IPv6머리부를 제외하고 옥테드단위로 준다.

이러한 파케트들에 대하여 IPv6머리부의 Payload Length마당은 령으로 설정되여야 하며 거기에는 Fragment머리부가 없다. 이 선택에 의해 IPv6의 4백만개 옥테드이상까지의 파케트크기를 지원한다. 이것은 큰 비데오파케트들의 전송을 쉽게 해주며 따라서 IPv6이 임의의 전송매체를 매우 잘 리용할수 있게 해준다.

Routing header는 파케트의 목적지로 가는 도중에 들리게 되는 하나 또는 그이상의 중간마디들의 목록을 포함한다. 모든 경로조종머리부들은 주어 진 경로조종형에 고유한 자료를 경로조종하여 잇닿은 4개의 8-bit마당들로 구성되는 32-bit블로크로 시작한다. 4개의 8-bit마당들로는 Next Header, Header Extension Length 그외 다음과 같은 2개의 마당이 있다.

- **Routing Type:** 특정한 Routing머리부를 식별한다. 만일 경로조종기가 Routing Type값을 인식하지 못하면 그 파케트를 칠회할수 있다.
- **Segments Left:** 마지막목적지에 도달하기전에 들리게 되는 중간마디들의 수

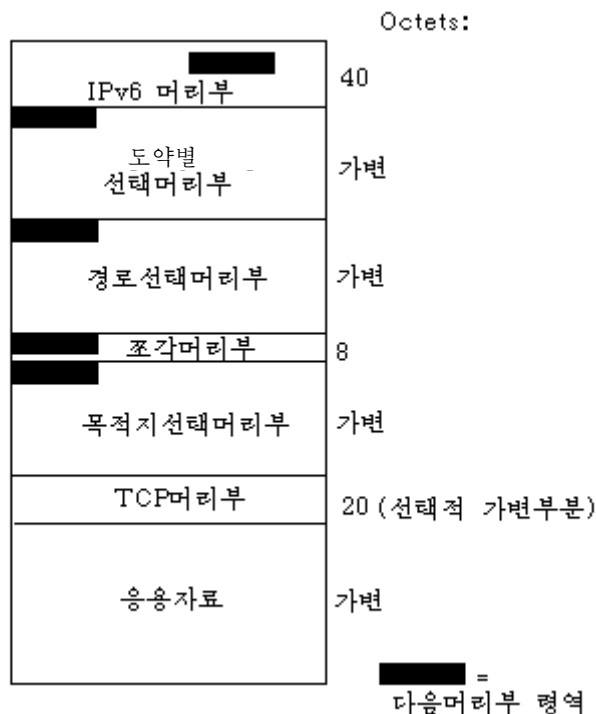



그림 13-15. 확장머리부들을 가지는 IPv6파케트

이 일반 머리부정의외에 IPv6 명세서는 Type 0 Routing머리부를 정의한다. Type 0 Routing머리부를 리용할 때 원천마디는 IPv6머리부에 마지막목적지의 주소를 주지 않는다. 대신 그 주소는 Routing머리부에 표기되는 마지막주소이며 IPv6머리부는 경로에서 처음 요구한 경로조종기의 목적지주소를 포함한다. Routing머리부는 그 파케트가 IPv6머리부에서 확인된 마디에 도달할 때까지 검사하지 않는다. 그 시험에서 IPv6과 Routing머리부내용들은 갱신되며 그 파케트로 전송된다. 갱신은 IPv6머리부에 들리게 될 다음주소를 두고 Routing머리부에 Segment Left마당을 감소시키는것으로 이루어 진다.

IPv6은 송신자에게 파케트를 돌려 주기 위하여 Routing머리부를 포함하여 반대의 경로들에 대한 IPv6마디를 요구한다.

Fragment header는 쪼각화가 요구될 때 원천지에서 리용한다. IPv6에서 쪼각화는 원천마디들에 의해서만 진행되며 파케트의 배달경로에 따르는 경로조종기들에 의해서는 진행되지 않는다. 호상련결망환경을 충분히 리용하기 위하여 마디는 경로우의 임의의 부분망에 의해 지원되는 제일 작은 최대전송단위(MTU)를 알수 있게 하는 경로발견알고리듬을 실행하여야 한다. 다시말하여 경로발견알고리듬은 마디가 경로우의 《병모가지》부분망의 MTU를 알수 있게 한다. 이 지식에 기초하여 원천마디는 매개 주어 진 목적지주소에 대하여 요구되는대로 쪼각화한다. 다시 말하여 원천은 모든 파케트들을 매개 부분망에 의해 지원되는 최소MTU인 1280octet들로 제한한다.

Next Header마당외에 쪼각머리부는 다음의 마당들을 포함한다.

- **Fragment Offset(13bits)**: 원래의 파케트에서 이 쪼각의 통신부하쪼각의 속한 곳을 지정한다.그것은 64bit단위로 측정된다. 이것은 쪼각(마지막쪼각이 아닌)등이 길이가 64bit의 배수인 자료마당을 포함하여야 한다는것을 의미한다.
- **Res(2bits)**: 앞으로의 리용을 위하여 예약
- **M Flag(1bit)**: 1=이상의 쪼각들; 0=마지막쪼각
- **Identification(32bits)**: 원래의 파케트를 유일적으로 식별하기 위해 요구한다. 식별자는 그 파케트가 인터네트에 남아 있는동안 파케트의 원천지주소와 목적지주소에 대하여 유일해야 한다. 같은 식별자, 원천지주소, 목적지주소를 가지는 모든 쪼각들도 재포함되여 원래의 파케트를 형성한다.

Destination Options header는 만일 존재하면 그 파케트의 목적지마디에 의해서만 조사되는 선택정보를 나른다. 이 머리부의 형식은 도약별 선택머리부와 같다.

제14장. Web보안

모든 기업들, 대부분 정부기관들과 많은 사람들이 자기의 Web싸이트들을 가지고 있다.인터네트에 접속하는 사람들과 회사들의 수는 급속히 늘어 나고 있으며 이들모두는 화상Web열람기를 가지고 있다. 기업들은 전자상거래를 위하여 Web우에 수단들을 설치하는데 열을 올리고 있다. 그러나 현실적으로 인터네트와 Web는 여러가지 공격에 대하여 매우 취약하다. 이러한 현실에 대처하여 폭 넓으며 안전한 Web봉사를 요구하는 목소리가 높아 지고 있다. Web보안문제의 범위는 대단히 넓으며 한권의 책에서 다 취급할수는 없다(이 장의 마지막에 그에 대한 일부 참고문헌들을 소개한다).

이 장에서는 Web보안을 위한 일반적인 요구들을 론의하고 다음 Web상업분야에서 점차 중요시되는 표준적인 방식들인 SSL/TLLS와 SET에 중점을 둔다.

14.1 Web보안의 필요성

WWW(World Wide Web)는 기본적으로 인터네트와 TCP/IP 인터네트상에서 동작하는 의뢰기/봉사기형의 응용이다. 그런것만큼 지금까지 이 책에서 서술한 보안도구들과 수법들은 Web보안문제와 관련된다. 그러나 문헌[GARF 97]에서 지적된것처럼 Web는 콤퓨터 및 망보안에 관하여 일반적으로 제기되지 않았던 새로운 문제들을 제기한다. 즉

- 인터네트는 쌍방향이다. 전통적인 출판환경들과는 달리 텔레텍스트(teletext), 음성응답(voice response) 또는 팍스백크(fax-back)를 포함하는 전자출판체계조차 인터네트상에서의 Web봉사기들에 대한 공격에 대하여 약하다.
- Web는 회사의 판로와 생산정보, 업무처리를 위한 가동환경으로 되고 있다.
- Web봉사기들이 파괴되면 평판이 나빠 질수 있으며 재정적으로 손해를 볼수 있다.
- Web열람기들이 매우 사용하기 쉽고 Web봉사기들의 구성과 관리가 비교적 쉬우며 Web내용이 점점 개발하기 쉬워 진다고 할지라도 그것이 기초하고 있는 쏘프트웨어는 매우 복잡하다. 이 복잡한 쏘프트웨어는 많은 보안상 결함들을 가지고 있을수 있다. Web의 짧은 력사에는 새롭고 갱신된 체계들이 여러가지 공격에 피해를 입은 실례들이 많다.
- Web봉사기는 주식회사나 정부의 콤퓨터체계에서 필수구성부분으로 되고 있다. Web봉사기의 보안이 파괴되면 공격자는 Web 그자체의 부분이 아니라 국부싸이트의 봉사기에 접속된 자료나 체계에 대한 접근을 얻을수 있다.
- Web봉사를 리용하는 사람들은 보안에 대하여 특별한 지식을 가지고 있지 않는 일반사람들이다. 이러한 사용자들은 현존하는 보안위험을 충분히 알고 있지 못하며 효과적인 대책들을 취할수 있는 도구들이나 지식을 가지고 있지 못하다.

Web보안위협

표 14-1에 Web를 리용하는데서 제기되는 보안위협들의 형태들에 대하여 제시하였

다. 이 위협들을 분류하는 한가지 방법은 능동과 수동공격으로 가르는것이다. 수동공격은 열람기들과 봉사기들사이의 망전송에 대한 도청과 Web싸이트상의 정보에 대한 접은을 포함한다. 능동공격에는 봉사기와 의뢰기사이의 전송에서 통보문들을 변경하는것, Web싸이트에서의 정보를 변경하는것 그리고 다른 사용자로 가장하는것 등이 속한다.

Web보안위협을 분류하는 또 다른 방법은 위협의 위치 즉 Web봉사기, Web열람기 그리고 열람기와 봉사기사이의 망전송에 관한것이다. 봉사기와 열람기보안문제는 콤퓨터체계보호의 범위에 속한다. 이 책의 4편에서는 일반콤퓨터체계의 보호뿐아니라 Web체계보호에 적절한 문제들을 취급한다. 전송보안문제들은 망보안범위에 속하므로 이 장에서 취급한다.

Web전송보안수법

Web보안을 제공하는 많은 수법들이 있다. 이미 고찰하였던 여러가지 방식들은 그것들이 제공하는 봉사, 그것들이 리용하는 방식에서 어느 정도 비슷하지만 그것들의 응용범위와 TCP/IP규약모임(protocol stack)안에서의 상대적인 위치에서는 다르다.

그림 14-1은 그 차이를 보여 준다. Web보안을 제공하는 한가지 방법은 IP보안(그림 14-1의 ㄱ)을 사용하는것이다. IPSec를 리용하면 그것이 말단사용자와 응용들에 대하여 투명하고 일반목적의 해결책을 제공하는 유리한 점이 있다. 더우기 IPSec는 선택된 전송만이 IPSec처리의 간접비용을 발생하도록 려파하는 능력을 가지고 있다.

표 14-1. Web에서 위협들의 비교[RUB197]

	위협	결과	대책
완정성	• 사용자자료의 변경 • 트로이목마열람기 • 전송중의 통보문변경 • 기억기의 변경	• 정보의 루실 • 기계의 손상 • 다른 모든 위협들에 대한 취약성	암호검사합
기밀성	• 망우에서의 도청 • 봉사기로부터 정보의 절취 • 망설정에 대한 정보 • 의뢰기가 봉사기에 보내는 정보	• 정보의 루실 • 개인성의 루실	암호, Web대행체들
봉사기의 거부	• 사용자스레드(Thread)들을 절단한다 • 기계에 필요 없는 정보들이 범람하게 한다 • 디스크나 기억기를 넘치게 한다 • DNS의 공격에 의한 기계의 격리	• 파괴적이다 • 사용자를 괴롭힌다 • 사용자가 작업을 진행하지 못하게 한다	예방이 어렵다
인증	• 정당한 사용자로의 가장 • 자료위조	• 부정사용자의 허가 • 거짓자료를 정당한 것으로 믿게 한다	암호화기술

또 다른 일반적인 해결책은 TCP의 바로 웃층에서 보안을 실현하는것이다(그림 14-1의 ㄴ). 이 수법의 가장 일반적인 실례는 SSL(Secure Sockets Layer)와 TLS (Transport Layer Security)로 알려 진 SSL의 인터네트표준이다. 이 준위에서 둘중 한가지 실현방법을 택할수 있다. 일반적으로 SSL(또는 TLS)은 기초규약모임의 한 부분으로 제공될수 있으며 결과 응용들에 대하여 투명해 진다. 또한 SSL은 개별적파케트들에 매몰시킬수 있다. 실례로 네트스케이프(Netscape)와 Microsoft Explorer열람기들은 SSL을 장비하고 있으며 대다수 Web봉사기들도 마찬가지이다.

응용고유의 보안은 특정한 응용프로그람내에서만 봉사한다. 그림 14-1의 ㄷ에 이 방식의 실례들을 보여 주었다.

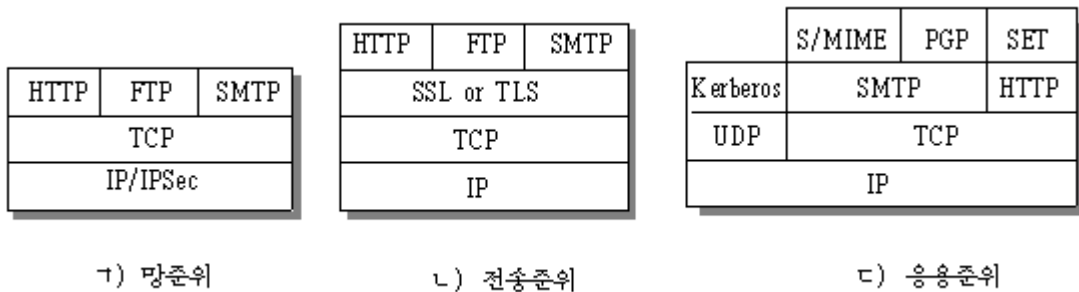



그림 14-1. TCP/IP규약보안수단들의 상대적인 위치

이 수법의 우월성은 봉사가 주어 진 응용의 특정한 요구들에 알맞게 제공된다는것이다. Web보안에서 이 수법의 전형적인 실례는 SET(Secure Electronic Transaction)이다.

이 장의 나머지부분에서는 SSL/TLS와 SET에 대하여 론의한다

14.2 안전소케트층과 전송층보안

SSL은 Netscape로부터 유래되였다. SSL의 판본 3은 일반관점에서 설계되여 인터네트초안문서로 공개되였다. 후에 이 규약을 인터네트규격으로 할데 대한 합의가 이루어졌으며 IETF내에 일반표준을 개발하기 위한 TLS연구그룹이 조직되였다. TLS에 대한 현재의 연구는 인터네트표준으로서의 판본을 구축하는데로 지향되고 있다. TLS의 첫 판본은 본질상 SSLv3.1로 볼수 있으며 SSLv3과 호환성을 가진다. 이 절은 대부분 SSLv3에 대하여 취급한다. 절의 마지막부분에서는 SSLv3과 TLS사이의 기본 차이점들을 서술한다.

SSL방식

SSL은 TCP를 리용하여 안전한 말단 대 말단봉사를 제공하도록 설계되였다.

SSL은 단일규약이 아니라 그림 14-2에서 보여 준것처럼 2층규약이다.

SSL레코드규약은 여러가지 보다 웃층의 규약들에 기초적인 보안봉사들을 제공한다. 특히 Web의 의뢰기/봉사기대화를 위한 전송봉사를 제공하는 HTTP는 SSL상에서 동작할수 있다.

보다 웃층의 세개의 규약들 즉 핸드쉐이크(Handshake)규약, 암호명세서변경(Change Cipher Spec)규약과 경고(Alert)규약이 SSL의 부분으로 정의된다. 이러한 SSL고유의 규약들은 SSL교환(exchange)들을 관리하는데 리용되며 이 절의 뒤부분에서 설명된다.

그림 14-2. SSL규약모임

두가지 중요한 SSL개념들은 SSL대화와 SSL접속인데 그것들은 명세서에서 다음과 같이 정의된다.

- **접속:** 접속이란 적당한 형태의 봉사를 제공하는(OSI참조모형정의에서) 전송이다. SSL에 대하여 이러한 접속들은 단과 단사이(peer to peer)관계들이다. SSL에 대하여 이러한 접속들은 일시적이다. 매개 접속들은 하나의 대화에만 관련된다.
- **대화:** SSL대화란 의뢰기와 봉사기사이의 관련(association)이다. 대화들은 핸드쉐이크규약에 의하여 창조된다. 대화들은 한 조의 암호보안파라메터들을 정의하는데 그것들을 여러개의 접속들에서 공유할수 있다. 매개 접속들을 위한 새로운 보안파라메터들을 류통하는데 드는 비용을 줄이기 위하여 대화를 리용한다.

임의의 대방들(의뢰기와 봉사기우에서 HTTP와 같은 응용들)사이에 여러가지 안전한 접속들이 있을수 있다. 리론적으로는 대방들사이에 여러가지 동시적인 대화들이 있을수 있으나 현실적으로 이러한 대화형태는 리용되지 않는다.

실제로 매 대화와 련관된 여러가지 상태들이 있다. 일단 대화가 확립되면 읽기와 쓰기(즉 받기와 보내기)들이 모두 가능한 상태로 된다. 또한 핸드쉐이크규약기간에 미정의 읽기와 쓰기상태들이 창조된다. 핸드쉐이크규약의 성공적인 결과에 의해 미정의 상태들이 이동상태들로 된다. 대화상태는 다음의 파라메터(SSL서술로부터 정의)들에 의하여 정의된다.

- **대화식별자:** 능동 또는 회복할수 있는 대화상태를 판별하기 위하여 봉사기가 선택하는 임의의 바이트렬이다.
- **단증명서:** 단(peer)의 X509.v3증명서이다. 이 요소의 상태는 무효로 될수 있다.
- **압축방법:** 암호화에 앞서 자료를 압축하는데 리용되는 알고리듬
- **암호명세서:** 모든 자료암호알고리듬(무효,DES 등과 같은)을 렬거하고 MAC계산에 리용되는 하쉬알고리듬(MD5 혹은 SHA-1과 같은것)을 서술한다.

- **주비밀열쇠:** 의뢰기와 봉사기사이에 공유된 48byte의 비밀열쇠
- **계속가능기발:** 그 대화를 새로운 접속들을 만드는데 리용할수 있는가 없는가를 가리키는 기발

접속상태는 다음의 파라메터들에 의하여 정의된다.

- **봉사기와 의뢰기의 우연수:** 매 접속을 위하여 봉사기와 의뢰기가 선택하는 바이트렬
- **봉사기쓰기MAC비밀열쇠(Server write MAC Secret):** 봉사기가 보낸 자료에 대한 MAC조작에 리용되는 비밀열쇠
- **의뢰기쓰기MAC비밀열쇠(Client write MAC secret):** 의뢰기가 보내온 자료에 대한 MAC조작에 리용되는 비밀열쇠
- **봉사기쓰기열쇠(Server Write Key):** 봉사기에 의해 암호화되고 의뢰기에 의해 복호되는 자료에 대한 전통암호열쇠
- **의뢰기쓰기열쇠(Client Write Key):** 의뢰기가 암호화하고 봉사기가 복호하는 자료에 대한 전통암호열쇠
- **초기화벡토르(IV):** CBC방식에서 블로크암호가 사용되면 벡토르초기화는 매 열쇠에 대하여 유지된다. 이 마당은 SSL의 핸드쉐이크규약에 의하여 처음에 초기화된다. 그로부터 매 레코드의 마지막암호본문블로크는 다음의 레코드의 초기화벡토르 IV로서의 리용을 위하여 보호된다.
- **렬번호(Sequence number):** 매 대방은 매 접속에 대한 통보문을 보내고 접속하기 위한 개별적렬번호들을 보관한다. 대방이 암호변경통보문을 보내거나 받을 때 해당한 렬번호는 0으로 설정된다. 렬번호들은 2^{64}-1을 넘지 않는다.

SSL레코드규약

SSL레코드규약은 SSL접속을 위하여 두가지 봉사를 제공한다.

- **기밀성:** 핸드쉐이크규약은 SSL통신부하의 전통암호에 리용되는 공유된 비밀열쇠를 정의한다.
- **통보문완정성:** 핸드쉐이크규약은 통보문인증코드(MAC)를 구성하는데 쓰이는 공유된 비밀열쇠를 정의한다.

그림 14-3에 SSL레코드규약의 전반적인 조작과정을 보여 주었다. 레코드통신규약은 전송되는 응용통보문을 다룰수 있는 블로크들로 쪼개고 선택적으로 자료를 압축하며 MAC를 적용하여 암호화한 다음 머리부를 더하고 그 결과단위를 TCP토막으로 전송한다. 수신된 자료는 먼저 복호되고 다음 검증되며 압축을 풀고 재조합한 다음 웃준위사용자에게 배달된다.

첫 단계는 **쪼각화**이다. 매 웃층의 통보문은 2^{14}byte(16384byte) 혹은 그이하의 블로크로 분할된다. 다음 **압축**이 선택적으로 적용된다. 압축과정은 루실이 없어야 하며 내용의 길이가 1024byte이상 증가하지 말아야 한다. SSLv3(TLS의 현재의 판본과 마찬가지로)에서는 그 어떤 압축알고리듬도 서술되여 있지 않기때문에 기정의 압축알고리듬은 무효이다.

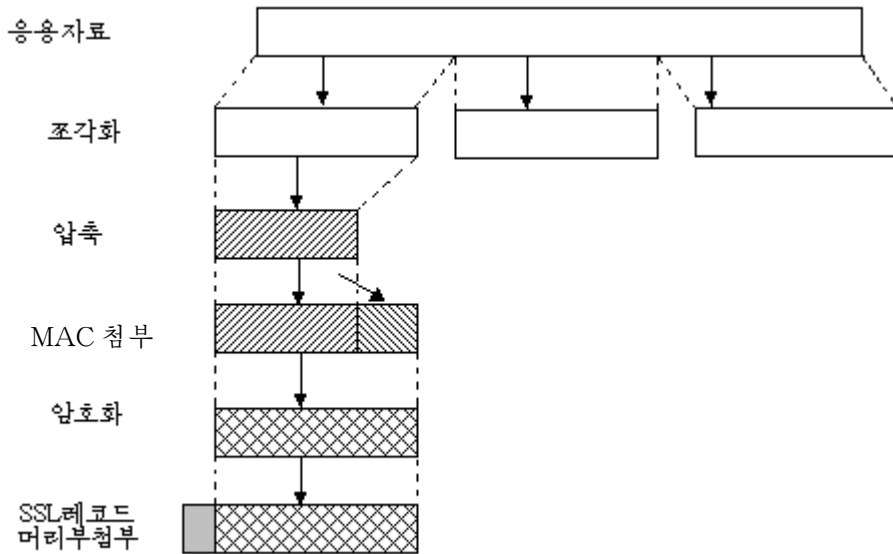



그림 14-3. SSL레코드규약조작

처리에서 다음단계는 압축한 자료에 대하여 **통보문인증코드(MAC)**를 계산하는것이다. 이를 위해서 공유된 비밀열쇠를 리용한다. 계산은 다음과 같이 정의된다.

hash(MAC_write_secret ‖ pad_2‖
hash(MAC_write_secret ‖ pad_1 ‖ seq_num‖SSLC Compressed.type ‖
SSLCompressed.length ‖ SSLCompressed.fragment))

여기서

‖	=	련결
MAC_write_secret	=	공유된 비밀열쇠
Hash	=	암호학적하쉬알고리듬; MD5 혹은 SHA-1
pad_1	=	바이트 0x36(0011 0110)는 MD5에 대하여 48번, (384bit) SHA-1에 대해서는 40번(320bit)반복
pad_2	=	바이트 0x5C(0101 100)은 MD5에 대하여 48번, SHA-1에 대하여 40번 반복
seq_num	=	이 통보문들에 대한 렬번호
SSLCompressed.type	=	이 통보문을 처리하는데 리용되는 웃준위규약
SSLCompressed.length	=	압축된 부분의 길이
SSLCompressed.fragment	=	압축된 조각(압축이 리용되지 않는다면 평문부분)

이것은 9장에서 정의한 HMAC알고리듬과 매우 류사하다. 차이점은 두개의 메구기들이 SSLv3에서는 련결되고 HMAC에서는 XOR된다는것이다. SSLv3 MAC알고리듬

은 련쇄(련결)를 리용한 HMAC를 위한 초기의 호상련결망초안에 기초하고 있다. RFC에서 정의된 HMAC의 마지막판본은 XOR를 리용한다. 다음 압축된 통보문과 MAC는 대칭암호를 리용하여 암호화되며 따라서 암호문은 총 길이가 2^{14}+2048을 넘지 않도록 그 내용의 길이가 1024byte이상 증가하지 말아야 하며 결과 다음의 암호알고리듬이 허락된다.

블로크암호		흐름암호	
알고리듬	열쇠크기	알고리듬	열쇠크기
IDEA	128	RC4-40	40
RC2-40	40	RC4-128	128
DES-40	40		
DES	56		
3DES	168		
Fortezza	80		

Fortezza는 맵시 있는 카드암호구성에 리용된다. 흐름암호로 서술된 압축통보문에 MAC를 덧붙여 암호화된다. MAC는 압축이 진행되기전에 계산되며 평문 또는 압축된 평문과 함께 암호화된다.

블로크암호화에서 메꾸기는 암호화하기전에 MAC다음에 부가된다. 메꾸기는 그 길이의 1byte표식이 붙는 많은 메꾸기바이트들의 형식으로 있다. 메꾸기의 총량은 암호화할 자료의 총 크기(평문+MAC+메꾸기)가 암호문블로크길이의 배수인 가장 작은 량이다. 실례는 SHA-1을 리용한 20byte의 MAC를 가진 58byte의 평문(또는 압축을 리용하면 압축한 본문)이며 그것은 8byte의 블로크길이를 리용하는 암호로 암호화된다(실례로 DES).

메꾸기길이를 포함하여 이것은 79byte를 차지한다. 전체를 8의 옹근배수로 하기 위하여 한 바이트의 메꾸기를 첨가한다.

SSL레코드규약처리의 마지막단계는 다음의 마당들로 이루어 지는 머리부를 준비하는것이다.

- **내용형(8bit):** 포함된 쪼각을 처리하는데 리용되는 웃층의 규약
- **기본판본(8bit):** 사용에서 SSL의 기본판본을 표시한다. SSLv3에 대한 값은 3이다.
- **낮은 판본(8bit):** 사용에서 더 낮은 판본(minor)을 표시.SSLv3에 대하여 그 값은 0이다.
- **압축한 길이(16bit):** 평문쪼각의 바이트길이(혹은 압축이 리용되면 압축한 쪼각)의 최대값은 2^{14}+2048이다.

정의된 내용형들은 change_cipher_spec, alert, handshake, application_data이다. 처음의 세개는 다음번에 론의하게 될 SSL고유의 규약이다. SSL을 리용할수 있는 여러가지 응용들사이에는 아무런 차이도 없으며 이러한 응용들에 의해 창조된 자료의 내용은 SSL에 대하여 불투명하다.

그림 14-4에 SSL레코드형식화를 보여 주었다.

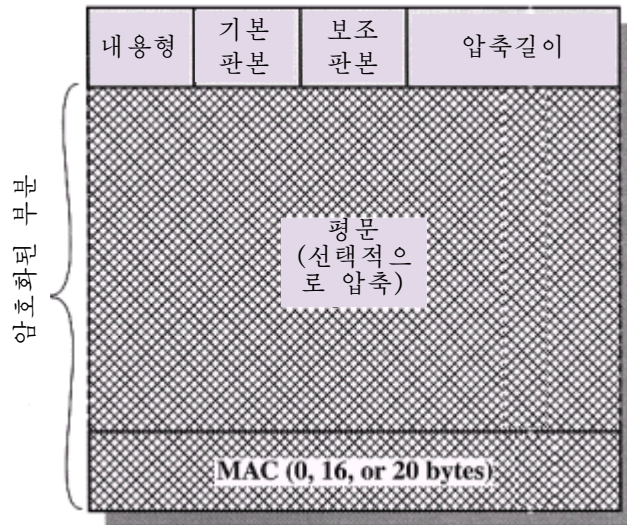



그림 14-4. SSL레코드형식화

암호명세서변경규약

암호명세서변경(Cipher Spec Change)규약은 SSL레코드규약을 리용하는 3개의 SSL고유의 규약들중에서 제일 간단한것이다. 이 규약은 하나의 통보문(그림 14-5의 ㄱ)으로 되는데 그것은 값 1을 가지는 하나의 바이트로 이루어 진다. 이 통보문의 유일한 목적은 미정의 상태를 현상태에로 복사시키는것인데 그과정에 이 접속에서 리용할수 있도록 암호묶음이 갱신된다.

경고규약

경고(Alert)규약은 SSL관련의 경고들을 단의 실체에로 전송하는데 리용된다. SSL을 리용하는 다른 응용들과 마찬가지로 경고통보문은 현상태에 의해 렬거된대로 압축되고 암호화된다.

이 규약의 매 통보문은 두 바이트로 이루어 진다(그림 14-5의 ㄴ). 첫 바이트는 값 warning(1) 또는 fatal(2)를 가지고 통보문의 중요도를 전달한다. 준위가 만일 fatal이라면 SSL은 접속을 즉시에 끝 마친다. 같은 대화에서 다른 접속들은 계속될수 있어도 이 대화에서 새로운 접속들은 설정할수 없다. 두번째 바이트는 특수경고를 표시하는 코드를 포함한다. 먼저 항상 중요한(SSL명세서에서 정의된것처럼) 경고들을 소개한다.

- unexpected_message: 타당치 못한 통보문이 접수되였다.
- bad_record_message: 정확하지 않은 MAC가 접수되였다.
- decompression_failure: 맞지 않는 입력을 접수하는 압축을 푸는 함수(즉 최대한 도의 허용길이보다 더 크게 압축을 풀거나 압축할수 없다).
- Handshake_failure: 송신자는 주어 진 선택에서는 보안파라메터를 정확히 설정할수 없다.

그림 14-5. SSL레코드규약통신부하

- illegal_parameter: 핸드쉐이크통보문에 있는 마당이 범위밖에 있거나 혹은 다른 마당들과 호환되지 않는다.

나머지경고들은 다음과 같다.

- close_notify: 이 접속에 대하여 송신자는 더이상 통보문을 보내지 않을것이라는것을 수신자에게 통보한다. 매 대방들은 접속의 쓰기측을 닫기전에 close_notify 경고를 보내야 한다.
- no_certificate: 정당한 증명서를 리용할수 없으면 증명서요구에 응답하는데 보내질수 있다.
- bad_certificate:접수한 증명서가 불결하다(실례로 검증되지 않는 서명을 포함).
- unsupported_certificate: 접수된 증명서의 형이 지원되지 않는다.
- certificate_revoked: 증명서가 그 서명자에 의하여 취소된다.
- certificate_expired: 증명서의 기한이 끝났다.
- certificate_unknown: 일부 다른 특기하지 않는 문제가 증명서를 처리하는데서 발생하였다.

핸드쉐이크규약(HANDSHAKE Protocol)

SSL의 제일 복잡한 부분은 핸드쉐이크규약(HANDSHAKE Protocol)에 있다.
이 규약은 의뢰기와 봉사기를 서로 인증하도록 하고 암호화,MAC알고리듬. SSL레코드에서 보낸 자료를 보호하기 위하여 리용되는 암호화열쇠들을 류통하도록 한다. 핸드쉐이크규약은 응용자료가 전송되기전에 리용된다.

- 핸드쉐이크규약은 의뢰기와 봉사기에 의하여 교환되는 련속적인 통보문들로 구성된다. 이것들은 모두 그림 14-5의 ㄷ에 보여 준 형식을 가진다. 매 통보문은 세개의 마당을 가진다.

표 14-2. SSL핸드쉐이크규약통보문형태

통보문형태	파라메터
hello_request:	무효
client_hello:	판본, 우연수, 대화식별자, 암호묶음, 압축방법
server_hello:	X.509v3증명서들의 련쇄
server_key_exchange:	파라메터, 서명
certificate_request:	형, 인증자
server_done	무효
certificate_verify	서명
client_key_exchange	파라메터, 서명
finished	하쉬값

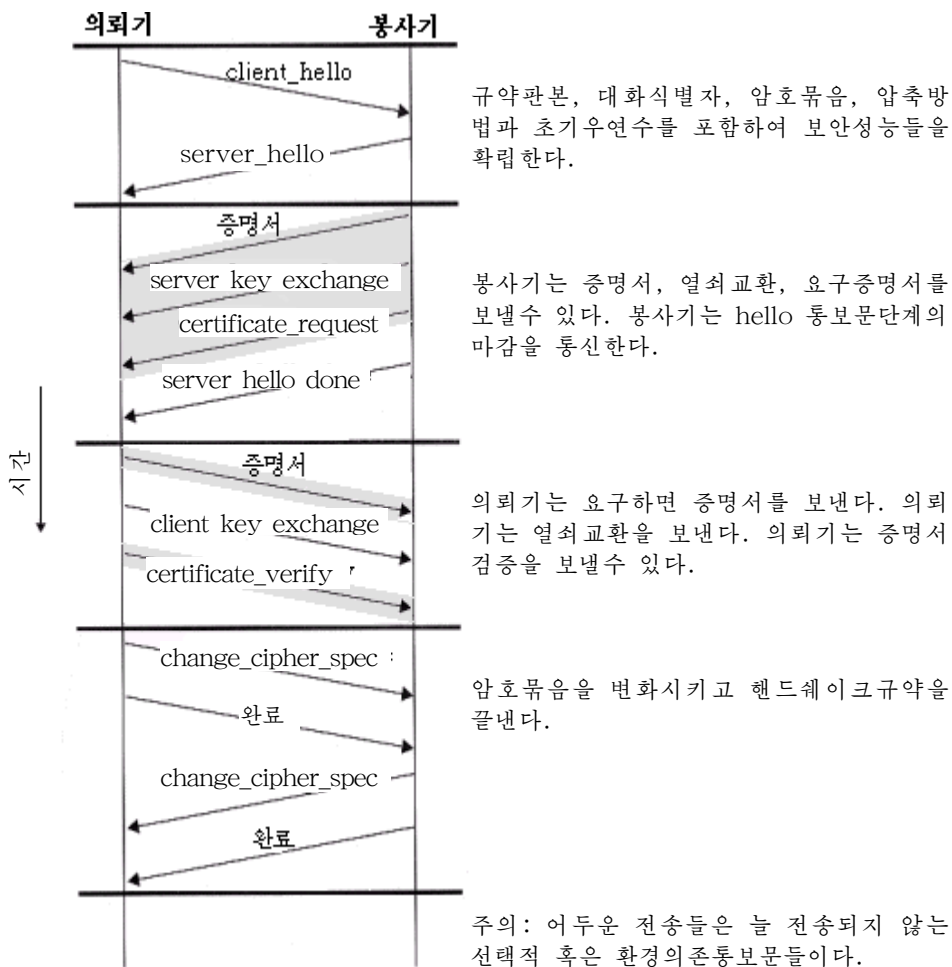



그림 14-6. 핸드쉐이크규약의 실행

- **형(1byte):**10개의 통보문들중의 하나를 표시한다. 표 14-2는 정의된 통보문형들을 보여 준다.
- **길이(3byte):**바이트단위의 통보문길이
- **내용(≥1byte):**파라메터들은 이 통보문과 관련된다. 이것들은 표 14-2에 보여 주었다.

그림 14-6에 의뢰기와 봉사기사이의 론리적인 접속을 설정하는데 필요한 초기교환을 보여 주었다. 교환은 4단계로 이루어 져 있다고 볼수 있다.

단계 1. 보안기능들을 설정

이 단계는 론리적인 접속을 시작하여 그와 관련한 보안기능들을 설정하는데 리용된다. 교환은 의뢰기에 의하여 시작되며 그것은 다음의 파라메터를 가진 client_hello_message를 보낸다.

- **판본:** 의뢰기에 있는 제일 높은 SSL판본
- **우연수:** 32bit의 시간표시와 안전한 우연수발생기에 의하여 발생된 28byte로 구성되는 우연구조. 이 값들은 림시값들로서 열쇠교환시 재연공격을 막기 위하여 리용된다.
- **대화ID:** 가변길이대화식별자. 령 아닌 값은 의뢰기가 이미 있는 접속의 파라메터를 갱신하거나 이 대화에 새로운 접속을 창조하려 한다는것을 표시한다.령값은 의뢰기가 새로운 대화에 새로운 접속을 확립하려는것을 표시한다.
- **암호묶음:** 이것은 의뢰기에 의하여 유지되는 암호화알고리듬의 결합(우선권준위가 작아 지는 차례로)을 포함하고 있는 목록이다. 이 목록의 매 암호묶음들은 열쇠교환알고리듬과 암호명세서를 둘 다 정의하며 이것들은 다음에 론의된다.
- **압축방법:** 이것은 의뢰기에 의하여 지원되는 압축방법들의 목록이다.

client_hello통보문을 보낸 다음 의뢰기는 server_hello통보문을 기다리는데 그것은 client_hello통보문과 같은 파라메터를 포함한다. server_hello통보문을 위하여 다음의 협약들이 적용된다. 판본마당은 의뢰기에 의하여 제기되는 더 낮은 판본을 포함하고 봉사기에 의하여 제기되는 제일 높은 판본을 포함한다. Random마당은 봉사기에 의하여 생성되며 의뢰기의 Random마당과는 독립이다. 만일 의뢰기의 Session ID마당이 령이 아니라면 봉사기는 같은 값을 리용한다. 그렇지 않으면 봉사기의 Session ID마당은 새로운 Session에 대한 값을 포함한다.Cipher Suite마당은 의뢰기에 의하여 정의된것들로부터 봉사기에 의하여 선택한 단 하나의 암호묶음을 포함한다. 압축마당은 의뢰기에 의하여 제의된것들에서 봉사기가 선택한 압축방법을 포함한다.

암호묶음파라메터의 첫 요소는 열쇠교환방법을 가리킨다(다시 말하여 전통암호열쇠와 MAC를 교환하는 방법). 다음의 열쇠교환방법들이 지원된다.

- **RSA:** 비밀열쇠는 수신자의 RSA공개열쇠로 암호화된다. 수신자의 열쇠에 대한 공개열쇠증명서를 반드시 리용하여야 한다.
- **고정디피-헬만:** 이것은 봉사기의 증명국(CA)이 서명한 디피-헬만공개파라메터를 포함하는 디피-헬만열쇠교환이다. 즉 공개열쇠증명서는 디피-헬만공개열쇠파라메터들을 포함한다. 의뢰기는 의뢰기검증이 요구되면 증명서에, 그렇지 않으면 열쇠교환통보문에 그것의 디피-헬만공개열쇠파라메터들을 제공한다.

- **림시디피-헬만**: 이 기술은 림시적인(Ephemeral) 비밀열쇠들을 창조하는데 리용된다. 이 경우에 송신자의 비밀RSA 또는 DSS열쇠를 리용하여 서명한 디피-헬만공개열쇠들을 교환한다. 수신자는 상응한 공개열쇠를 리용하여 서명을 검증할수 있다. 증명서들은 공개열쇠를 확증하는데 리용된다. 이것은 림시적인 인증열쇠를 생성하므로 3개의 디피-헬만선택들중에서 제일 안전할것이다.
- **닉명의 디피-헬만:** 기본디피-헬만알고리듬은 인증없이 리용된다. 즉 대방들은 자기의 공개디피-헬만파라메터들을 상대방에게 인증없이 보낸다. 이 방식은 쌍방과 닉명의 디피-헬만을 리용하는 제3자의 공격에는 약한 결함이 있다.
- **포테자(Fortezza):** 포테자방식에 대하여 정의된 기술

열쇠교환방법에 대한 정의는 다음의 마당을 포함하는 **암호명세서**이다.

- **암호알고리듬:** 이미 언급한 알고리듬들 RC4, RC2, DES, 3DES, DES40, IDEA, Fortezza중의 어느 하나이다.
- **MAC알고리듬:** MD5 또는 SHA-1
- **암호형**: 흐름 또는 블로크
- **반출가능형**: 참 혹은 거짓
- **하쉬크기:** 0,16byte(MD5에 대하여) 혹은 20byte(SHA-1에 대하여)
- **열쇠자료:** 쓰기열쇠들을 생생하는데 리용되는 자료를 포함하는 바이트렬
- **IV크기:** CBC암호화에서 초기값의 크기

단계 2. 봉사기인증과 열쇠교환

봉사기는 인증이 요구되면 그것의 증명서를 보내는것으로 이 단계를 시작한다. 즉 통보문은 하나의 X.509증명서 혹은 그것들의 련쇄를 포함한다. 증명서통보문은 닉명의 디피-헬만을 제외한 임의의 동의된 열쇠교환방법에 요구된다. 만일 고정디피-헬만이 사용되면 증명서통보문은 그것이 봉사기의 공개디피-헬만파라메터를 포함하기때문에 봉사기열쇠교환통보문으로 리용된다.

다음으로 server_key_exchange통보문은 그것이 요구될 때 전송된다. 그것은 다음의 두 경우에는 요구되지 않는다.

(1) 봉사기가 고정디피-헬만파라메터들을 가진 증명서를 보내거나

(2) RSA열쇠교환이 사용되는 경우

Server_key_exchange통보문은 다음의 경우에 요구된다.

- **닉명의 디피-헬만:** 통보문내용은 두개의 디피-헬만대역값들(씨수와 그의 원시뿌리)과 봉사기의 공개디피-헬만열쇠를 포함한다(그림 6-16).
- **림시디피-헬만:** 통보문내용은 닉명의 디피-헬만에 대하여 제공되는 세개의 디피-헬만파라메터들과 그 파라메터들에 대한 서명을 포함한다.
- **RSA열쇠교환-봉사기는 RSA를 리용하지만 서명전용RSA열쇠를 가지지 않는 경우 :** 따라서 의뢰기는 봉사기의 공개열쇠로 암호화된 비밀열쇠를 간단히 보낼수 없다. 대신 봉사기는 림시RSA공개/비밀열쇠쌍을 창조하고 server_key_exchange통보문을 리용하여 공개열쇠를 보내야 한다.통보문내용은 림시RSA공개열쇠(그림 6-5)의 두개의 파라메터와 그 파라메터들에 대한 서명을 포함한다.
- Fortezza

보통 서명은 통보문의 하쉬를 취하고 그것을 송신자의 공개열쇠로 암호화하여 생성된다. 이 경우에 하쉬는 hash(clientHello.random || serverHello.random || serverParams)로 정의된다. 따라서 하쉬는 디피-헬만이나 RSA파라메터들뿐아니라 초기의 hello통보문으로부터 두개의 한번쓰기정보들도 포함한다. 그로부터 재연공격과 허위진술로부터 안전하게 한다. DSS서명의 경우에 하쉬는 SHA-1알고리듬을 리용하여 진행된다. RSA서명인 경우에 MD5와 SHA-1하쉬는 둘 다 진행되며 두개의 하쉬(36byte)들의 련결은 봉사기의 공개열쇠로 암호화된다.

다음 닉명이 아닌 봉사기(닉명의 디피-헬만을 사용하지 않는 봉사기)는 의뢰기로부터 증명서를 요구할수 있다. **certificate_request**통보문은 두 파라메터 즉 certificate_type와 certificate_authority들을 포함한다. 증명서형은 공개열쇠알고리듬과 그 리용을 표시한다.

- RSA,서명전용
- DSS,서명전용
- 고정디피-헬만을 위한 RSA: 이 경우에 서명은 RSA로 서명된 증명서를 보내여 인증에 리용된다.
- 고정디피-헬만을 위한 DSS: 인증하는데만 재사용
- 림시디피-헬만용RSA
- 하루살이디피-헬만용DSS
- Fortezza

certificate_request통보문에서 두번째 파라메터는 신용되는 증명국들의 이름들을 구별하는 목록이다. 단계 2의 마지막통보문과 늘 요구되는 통보문은 **server_done**통보문인데 이것은 봉사기 hello와 관련통보문의 끝을 나타내기 위하여 봉사기가 보낸다. 이 통보문을 보낸후에 봉사기는 의뢰기응답을 기다린다. 이 통보문은 파라메터를 가지지 않는다.

단계 3. 의뢰기인증과 열쇠교환

sever_done통보문의 접수후 의뢰기는 요구되는 봉사기에 정당한 증명서가 제공되였는가를 확인하고 sever_hello파라메터들이 접수가능한가를 검사한다. 만일 모든것이 만족되면 의뢰기는 하나 혹은 그이상의 통보문을 봉사기에 다시 보낸다.

봉사기가 증명서를 요구하였으면 의뢰기는 **증명서통보문**을 보내는것으로 이 단계를 시작한다. 만일 알맞는 증명서를 쓸수 없으면 의뢰기는 대신 no_certificate alert를 보낸다.

다음은 **client_key_exchange**통보문인데 이것은 이 단계에서 반드시 보내야 하는것이다. 통보문의 내용은 열쇠교환의 형에 의존하는데 다음과 같다.

- **RSA**:의뢰기는 48byte의 *pre_master secret*를 생생하고 봉사기의 증명서에 있는 공개열쇠 또는 server_key_exchange통보문의 림시RSA열쇠로 암호화한다. *Master secret*를 계산해야 할 필요성에 대하여서는 후에 설명한다.
- **림시 혹은 닉명의 디피-헬만**: 의뢰기의 공개디피-헬만파라메터가 설정된다.
- **고정된 디피-헬만**: 의뢰기의 공개디피-헬만파라메터들이 증명서통보문으로 보내졌으므로 이 통보문의 내용은 무효이다.
- **Fortezza**: 의뢰기의 fortezza파라메터들을 보낸다.

마지막으로 이 단계에서 의뢰기는 의뢰기증명서의 정확한 확인을 제공하기 위하여 **certificate_verify**통보문을 보낼수 있다.

이 통보문은 다만 서명능력을 가진 다음의 어떤 의뢰기증명서만을 보낸다(실례로 고정디피-헬만파라메터들을 포함한것들을 제외한 모든 증명서들). 이 통보문은 아래와 같이 정의된 선행한 통보문에 기초한 하쉬코드를 가지고 서명된다.

```
Certificate Verify.signature.md5_hash
    MD5(master_secret || pad_2 || MD5(handshake_messages || master_secret ||
    pad_1 |));
Certificate.signature.sha_ hash
    SHA(master_secret || pad_2 || SHA(handshake_messages || master_secret
    || pad_1));
```

여기서 pad_1과 pad_2는 MAC에 대해 이미 정의된 값이며 핸드쉐이크통보문은 이 통보문을 포함하지 않지만 client_hello에서 시작을 보내고 받는 모든 핸드쉐이크규약을 참조하며 master_secret는 그것의 구조가 이 절에서 후에 설명하게 되는 계산된 비밀값이다. 만일 사용자의 비밀열쇠가 DSS이면 그것은 SHA-1하쉬를 암호화하는데 리용된다. 사용자의 비밀열쇠가 RSA이면 그것은 MD5와 SHA-1하쉬들의 련결을 암호화하는데 리용된다. 어느 경우에도 목적은 의뢰기증명서용의 비밀열쇠에 대한 의뢰기의 소유권을 증명하는것이다. 만일 어떤 사람이 그 의뢰기의 증명서를 악용한다고 해도 그는 이 통보문을 보낼수 없을것이다.

단계 4. 끝내기

이 단계는 안전한 접속의 설정을 완료한다. 의뢰기는 암호명세서변경통보문을 보내고 현재의 암호명세서에 미정의 암호명세서를 복사한다. 이 통보문이 암호명세서변경규약을 사용하여 보내온것이지만 핸드쉐이크규약의 부분을 고려하지 않았다는것을 주의해 둔다.

다음 의뢰기는 곧 새로운 알고리듬, 열쇠들과 비밀에 따라서 완료통보문을 보낸다. 완료통보문은 열쇠교환과 확증처리들이 성공적으로 되였는가를 확인한다. 완료통보문의 내용은 두 하쉬값

```
MD5(master_secret||pad_2||MD5(handshake_messages||Sender||master_secret
||pad_1));
SHA(master_secret||pad_2||SHA(handshake_messages||Sender||master_secre
t||pad_1));
```

등의 련결이다. 여기서 Sender는 의뢰기이고 핸드쉐이크통보문이 모든 핸드쉐이크통보문으로부터 이것을 포함하지 않는것까지의 모든 자료들이라는것을 확인하는 코드이다.

이 두 통보문에 대해 봉사기는 자기의 change_cipher_spec통보문을 보내고 현재의 CipherSpec에로 미정을 전송하여 자기의 완료통보문을 보낸다. 이 시점에서 핸드쉐이크는 완료되며 의뢰기와 봉사기는 응용층의 자료를 교환하기 시작한다.

암호화처리

두 조항 즉 열쇠교환으로 공유된 주비밀열쇠의 창조와 주비밀열쇠로부터 암호파라메터들의 생성이 중요하다.

주비밀열쇠의 창조

공유된 주비밀열쇠는 안전한 열쇠교환에 의해 이 대화을 위하여 생성된 1회용 48-byte(384bit)값이다. 생성은 두 단계에 걸쳐 진행된다. 먼저 pre_master_secret가 교환되고 다음 master_secret가 쌍방들에서 계산된다. Pre_master_secret교환에 대하여 두가지 가능성이 있다.

- **RSA**:48-byte pre_master_secret는 의뢰기에 의하여 생성되며 봉사기의 공개 RSA열쇠로 암호화되여 봉사기에 송신된다. 봉사기는 자기의 비밀열쇠로 암호문을 분석하여 pre_master_secret를 복호한다.
- **고정된 디피-헬만**:의뢰기와 봉사기는 둘 다 디피-헬만공개열쇠를 생성한다. 이것들이 교환된 다음 매측은 공유된 pre_master_secret를 창조하기 위하여 디피-헬만처리(계산)를 진행한다.

쌍방이 다 다음과 같이 master_secret를 계산한다.

```
master_secret = MD5(pre_master_secret || SHA('A' || pre_master_secret ||
                    ClientHello.random || ServerHello.random)) ||
                MD5(pre_master_secret || SHA('BB' || pre_master_secret ||
                    ClientHello.random || ServerHell.random)) ||
                MD5(pre_master_secret || SHA ||'CCC'|| pre_master_secret ||
                    clientHello. random || ServerHello.random))
```

여기서 ClientHello.random과 ServerHello.random은 초기hello통보문에서 교환되는 두개의 한번쓰기정보값들이다.

암호화파라메터의 생성

CpherSpec들은 의뢰기MAC작성비밀,봉사기MAC작성비밀, 의뢰기쓰기열쇠, 봉사기쓰기열쇠,의뢰기쓰기IV, 봉사기쓰기IV를 요구하는데 그것들은 이 순서로 주비밀열쇠로부터 생성된다. 이 파라메터들은 주비밀열쇠를 하쉬하고 모든 필요한 파라메터들을 위한 충분한 길이의 안전한 바이트렬을 만듦으로써 생성된다.

주비밀열쇠로부터 열쇠자료의 생성은 pre_master secret로부터 주비밀열쇠의 생성과 같은 절차로서 진행된다. 이것은 충분한 량의 결과가 얻어 질 때까지 계속된다.

```
Key_block = MD5(pre_master_secret || SHA('A'||master_secret ||
                 ServerHello.random || ClientHello.random))||
            MD5(pre_master_secret||SHA('BB'||master_secret||
                 ServerHello.random||ClientHello.random))||
            MD5(pre_master_secret||SHA||'CCC'||master_secret||
                 ServerHello.lrandom||ClientHello.random))...
```

이 알고리듬구는 준우연함수이다. 함수에 대한 준우연씨값을 master_secret로 볼수 있다. 의뢰기와 봉사기의 우연수들은 암호분석을 복잡하게 하는 염(salt)값으로 볼수 있다(salt값의 사용을 론의한 15장을 보시오).

전송층보안

TLS는 SSL의 인터네트표준을 만드는것을 목적으로 하는 IETF에서 제출한 표준규격이다. 현재 TLS의 초판은 SSLv3과 매우 류사하다. 이 절에서는 차이점들을 강조한다.

판본번호

TLS레코드형식은 SSL레코드형식과 같다(그림 14-4). 그리고 머리부의 마당들은 같은 의미를 가진다. 한가지 다른 점은 판본값에 있다. TLS의 현재4초안에 대하여 주요판본은 3이며 부차적인 판본은 1이다.

통보문인증코드

SSLv3과 TLS MAC방식사이에는 두가지 즉 실제의 알고리듬과 MAC처리(계산)의 범위에서 차이가 있다. TLS는 RFC 2104에 정의된 HMAC알고리듬을 리용한다. HMAC는 다음과 같이 정의된다.

$$HMAC_K = H[(K^+ \oplus opad) || H[(K^+ \oplus ipad) || M]]$$

여기서

H = 매몰된 하쉬함수(TLS에서 MD5나 SHA-1)

M = HMAC에 대한 통보문입력

K^+ = 결과가 하쉬코드의 블로크길이와 같도록 왼쪽에 령을 채워 넣은 비밀열쇠 (MD5와 SHA-1에서 블로크길이는 512bit)

ipad = 00110110(16진수로 36)이 64번(512bit) 반복된것

opad = 01011100(16진수로 5C)이 64번(512bit) 반복된것

SSLv3은 덧붙인 바이트들이 블로크길이에 덧붙인 비밀열쇠와 XOR되지 않고 비밀열쇠와 련결된것을 제외하고는 같은 알고리듬을 리용한다. 보안의 준위는 두 경우 대략 같다.

TLS에서는 다음의 식으로 표시되는 마당들을 MAC(계산)의 대상으로 한다.

HMAC_hash(MAC_write_secret, seq_num||TLSCompressed.type||
TLSCompressed.version||TLSCompressed.length||TLSCompressed.fragment))

MAC계산은 SSLv3계산에 포함된 모든 마당들과 리용하고 있는 규약의 판본인 TLSCompressed.Version마당을 포함한다.

준우연함수

TLS는 열쇠생성이나 정당성검증의 목적으로 PRF라는 우연함수를 리용하여 비밀열쇠들을 자료블로크에로 확장한다. 목적은 하쉬함수와 MAC에 대한 여러가지 공격에 대하여 안전한 방법으로 비교적 작은 공유된 비밀값을 리용하여 보다 긴 자료블로크들을 생성하는것이다. PRF는 다음의 자료확장함수에 기초한다(그림 14-7).

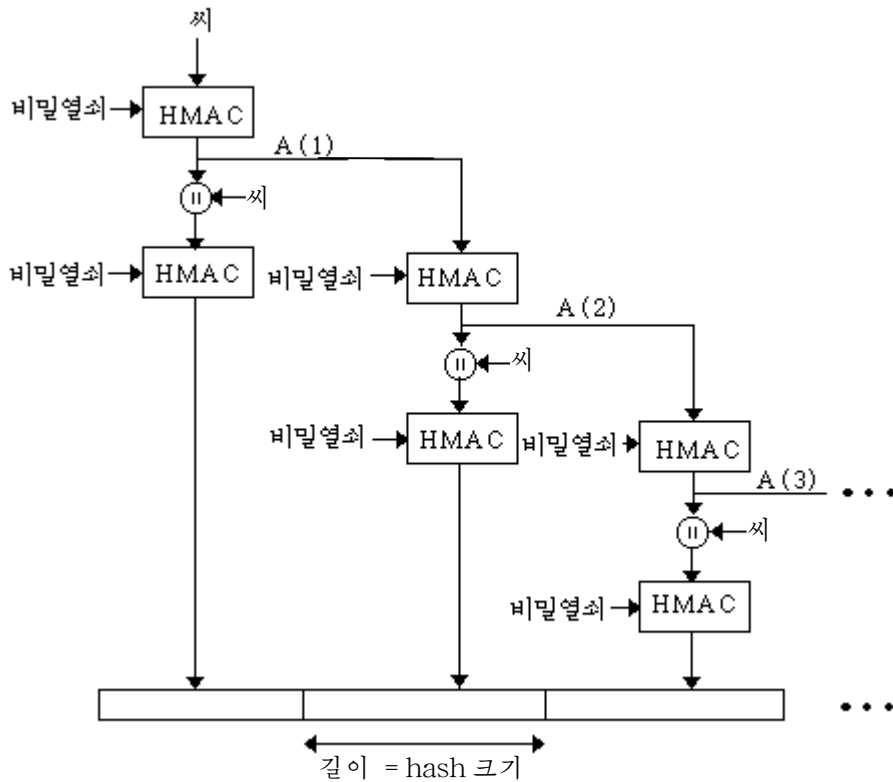



그림 14-7. TLS함수 P_하쉬(비밀값, 씨)

$$
\begin{aligned}
\text{P_hash(secret, seed)} = & \text{ HMAC_hash(secret, A(1)} \| \text{seed)} \| \\
& \text{ HMAC_hash(secret, A(2)} \| \text{seed)} \| \\
& \text{ HMAC_hash(secret, A(3)} \| \text{seed)} \| \cdots
\end{aligned}
$$

여기서 A()는 다음과 같이 정의된다.

$$
\begin{aligned}
& \text{A(0)=seed.} \\
& \text{A}(i)\text{=HMAC_hash(secret, A}(i-1))
\end{aligned}
$$

자료확장함수는 기본하쉬함수로서 MD5나 SHA-1들중 어느 하나를 가지는 HMAC 알고리듬을 리용한다. 보는바와 같이 P_hash는 요구되는 자료량을 만드는데 필요되는 회수만큼 반복될수 있다. 실례로 P_SHA-1를 64byte의 자료를 생성하는데 리용한다면 그것을 4번 반복하여야 80byte의 자료를 만들수 있다. 거기서 마지막16byte는 버린다. 이 경우에 P_MD5도 역시 4번 반복되여야 정확히 64byte의 자료를 만들수 있다. 매 반복마다 두번의 HMAC실행이 포함되는데 매 실행은 또 두번의 기본하쉬알고리듬의 실행을 포함한다.

PRF를 될수록 안전하게 하기 위하여 어느 알고리듬이든 안전하면 그것의 보안을 담보하는 방법으로 두개의 하쉬알고리듬을 리용한다. PRF는 다음과 같이 정의된다.

$$PRF(secret, lable, seed) = P_MD5(S1, label \| seed) \oplus P_SHA\text{-}1(S2, label \| seed)$$

PRF는 입력으로 비밀값, 식별표식과 씨값을 취하여 임의의 길이의 출력을 만든다. 비밀값을 절반(S1과 S2)으로 나누고 매 절반에 대하여 P_hash를 적용하고 하나의 절반에는 MD5를, 다른 절반에는 SHA를 리용하여 출력을 만든다. 그 두 결과들은 XOR되여 출력을 이룬다. 이를 위하여 P_MD5는 일반적으로 XOR함수에 대한 입력자료와 같은 량의 자료를 만들수 있다.

경고코드(Alert Codes)

TLS는 SSLv3에 정의된 no_certificate를 제외한 모든 경고코드들을 지원하고 있다. TLS에는 많은 코드들이 더 정의되여 있다. 이것들중 다음의것들은 치명적인 경고들이다.

- decryption_failed:타당치 않은 방법으로 복호된 암호문 즉 검사시에 그것이 블로크길이의 짝수배가 아니거나 그것의 덧붙임값들이 정확치 않다.
- record_overflow:길이가 2^{14}+2048byte를 넘는 통신부하이나 혹은 길이가 2^{14}+1024byte보다 긴 암호문을 가지는 TLS레코드가 접수되였다.
- unknown_ca: 정당한 증명서련쇄 혹은 부분련쇄를 접수하였지만 그 증명서를 CA로 확인할수 없거나 알려 진 믿을만한 CA와 일치되지 않기때문에 접수할수 없다.
- access_denied: 타당한 증명서가 수신되였지만 접근조종을 적용한 결과 송신자가 류통을 계속하지 않기로 하였다.
- decode_error: 마당이 자기의 정의범위를 벗어 나거나 통보문의 길이가 정확치 않기때문에 통보문은 복호할수 없다.
- export_restriction: 열쇠길이에 대한 전파제한에 따르지 않는 류통이 발견되였다.
- protocol_version: 의뢰기가 류통을 시도한 규약판본이 확인되였지만 지원되지 않는다.
- Insufficient_security: 봉사기가 의뢰기에 의하여 지원되는것보다 더 안전한 암호들을 요구하기때문에 류통이 실패할 때 handshake_failure를 대신하여 돌려 준다.
- Internal_error: 단 또는 규약의 정확성에 무관계한 내부오유는 통신을 계속할 수 없게 한다.

새로운 경고들중의 나머지는 다음과 같다.

- decrypt_error: 핸드쉐이크암호화조작은 서명을 검증할수 없고 열쇠교환을 복호화하거나 끝내기통보문을 유효하게 할수 없는것 등을 비롯하여 실패한다.
- user_concealed: 규약의 부족점과 관련되지 않는 일련의 리유로 하여 이 핸드쉐이크가 무효로 된다.
- no_renegotiation: 의뢰기에 의하여 hello요구에 응답을 보내거나 봉사기에 의하여 초기핸드쉐이크후에 의뢰기hello에 대한 응답을 보낸다. 이 통보문들중의 하나는 보통 재류통에 귀착되지만 이 정보는 송신자가 재류통을 할수 없다는것을 표시한다. 이 통보문은 항상 경고이다.

암호묶음(Cipher Suites)

SSLv3에서 리용할수 있는것과 TLS에서 리용할수 있는 암호묶음들사이에는 일부 약간한 차이들이 있다.

- **열쇠교환:** TLS는 Fortezza를 제외한 SSLv3의 모든 열쇠교환기술을 지원한다.
- **대칭암호화알고리듬:** TLS는 Fortezza를 제외한 SSLv3에 있는 모든 대칭암호알고리듬들을 지원한다.

의뢰기증명서형

TLS는 증명서요구통보문에서 요구되는 다음의 증명서들을 정의한다. rsa _sign, dss_sign, rsa_fixed_dh, dss_fixed_dh이다. 이것들은 모두 SSLv3에서 정의된다. 또한 SSLv3은 rsa_ephermeral_dh, dss_ephermeral_dh와 fortezza_kea들을 포함한다. 림시디피-헬만은 RSA 혹은 DSS들중의 하나를 가지고 디피-헬만파라메터의 서명을 포함한다. 즉 TLS에 대하여 rsa_sign과 dss_sign형들은 그 함수로 사용되며 개개의 서명형은 디피-헬만파라메터들을 서명하는데는 필요가 없다. TLS는 Fortezza방식을 포함하지 않는다.

Certificate_Verify와 완료통보문

TLS Certificate_Verify통보문에서 MD5와 SHA-1하쉬들은 다만 핸드쉐이크통보문우에서만 계산된다. SSLv3에 대하여 하쉬계산들 역시 주비밀열쇠와 pad들을 포함하였다는것을 상기하자. 이때문에 추가마당들은 보안을 더 부가하지 못한다는것을 알수 있다.

SSLv3에서의 완료통보문처럼 TLS에서도 완료통보문은 공유된 master_secret이전의 핸드쉐이크통보문들 및 의뢰기 또는 봉사기를 식별하는 표식에 기초한 하쉬이다. TLS에서 계산은 좀 다르다. 즉

PRF(master_secret, finished_lable, MD5(handshake_messages)||SHA-1(handshake_messages)이다.

여기서 finished_label은 의뢰기에 대하여 문자렬 《의뢰기완료》이고 봉사기에 대하여 《봉사기완료》이다.

암호화처리

TLS에서 pre_master_secret는 SSLv3에서와 같은 방법으로 계산된다.

SSLv3에서처럼 TLS에서도 master_secret는 pre_master_secret와 두개의 hello우연수들의 하쉬함수로서 계산된다.

TLS의 계산형식은 SSLv3와 다르며 다음과 같이 정의된다.

```
master_secret=
    PRF(pre_master_secret, "master secret", ClientHello.random||ServerHello.
    random)
```

알고리듬은 48byte의 준의연출력이 얻어 질 때까지 진행된다. 열쇠블로크자료(MAC비밀열쇠들, 대화암호화열쇠들과 IV)의 계산은 다음과 같이 정의된다.

```
key_block=
    PRF(master_secret, " key expansion",
      Security Parameters.server_random‖SecurityParameters.client_random)
```

SSLv3과 같이 열쇠블로크는 master_secret와 의뢰기 및 봉사기의 우연수들의 함수이다. 그러나 TLS에 대하여 실제적인 알고리듬은 다르다.

메꾸기

SSL에서 사용자자료의 암호화에 앞서 첨가되는 메꾸기(padding)는 암호화될 자료의 총 크기가 암호문블로크길이의 배수가 되도록 요구되는 최소한도의 량이다. TLS에서 메꾸기는 최대로 255byte까지 암호문블로크길이의 배수로 얻어 지는 임의의 량이 될수 있다. 만일 평문(혹은 압축이 요구되면 압축된 본문)과 MAC 그리고 메꾸기길이바이트가 모두 79byte이면 바이트단위로 메꾸기의 길이는 1,9,17 등 249까지 될수 있다. 여러가지 메꾸기 길이는 교환된 통보문의 길이해석에 기초한 공격을 좌절시키는데 리용할수 있다.

14.3 안전한 전자거래

안전한 전자거래(Secure Electronic Transaction:SET)은 인터네트에서 신용카드거래를 보호하기 위하여 설계되였으며 공개된 암호와 보안을 위한 명세서이다. 1996년 2월에 MasterCard와 Visa에 의한 보안표준에 대한 요구로부터 현재의 판본 SETv1이 출현하였다. Microsoft, Netscape, RSA, Terisa와 Verisign들을 비롯하여 많은 회사들이 최초의 설계명세서를 개발하는데 열중하였다. 1996년 초에 수많은 안들이 시험되였으며 1998년에 첫 SET_compliant제품들이 나오게 되였다.

SET 그자체는 지불체계가 아니라 사용자들이 신용카드결재의 하부구조를 인터네트와 같은 열린 망에서 안전하게 실현하기 위한 보안규약과 형식들의 모임이다.

본질적으로 SET는 다음의 세가지 봉사를 제공한다.

- 해당 거래에 관계하는 모든 대방들에 안전한 통신통로를 제공한다.
- X.509v3수자증명서에 기초한 신용을 제공한다.
- 정보가 필요한 때, 필요한 곳에서 거래의 대방들만 쓸수 있으므로 개인성을 담보한다.

SET는 1997년 5월에 발행된 세권의 책에서 정의된 복잡한 명세서이다.

- 1권:업무해설(80페지)
- 2권:프로그람작성지도서(629페지)
- 3권:형식적규약명세서(262페지)

이것은 총 971페지의 설명서이다. 반대로 SSLv3설명서는 63페지이며 TLS설명서는 71페지이다. 따라서 이 많은 설명서의 개요만을 이 절에서 서술한다.

SET 개괄

SET에 대한 론의를 SET에 대한 업무요구, 그것의 기본특성들과 SET거래의 관계자들을 고찰하는것으로부터 시작할수 있다.

요구

SET명세서의 1권은 인터네트와 다른 망들에서 신용카드에 의한 안전한 지불처리에

대한 다음의 사무요구를 서술한다.

- **결재와 주문정보의 기밀성을 제공할것:** 이 정보가 안전하고 지정된 접수자만이 볼수 있다는것을 신용카드소유자들에게 확신시키는것이 필요하다. 또한 기밀성은 거래대방이나 제3자에 의한 기만에 의한 위험을 줄인다. SET는 기밀성을 보장하는데 암호를 사용한다.
- **전송된 모든 자료의 완정성을 담보할것:** 즉 SET통보문들이 전송기간에 내용에서 아무런 변화도 없다는것을 담보한다. 완정성을 제공하는데 수자서명을 리용한다.
- **카드소유자가 신용카드구좌의 정당한 사용자라는것을 인증할것:** 개별적구좌번호에 카드소유자를 련결시키는 기구에 의해 협잡과 지불처리의 전반적인 비용을 줄인다. 수자서명과 증명서들은 카드소유자가 해당구좌의 정당한 사용자이라는것을 증명하는데 리용된다.
- **판매점이 재정기관과의 련계를 통하여 신용카드거래를 접수할수 있다는 인증을 제공할것:** 이것은 우의 요구에 대한 보충이다. 카드소유자는 자기들이 안전한 결재를 처리할수 있는 판매점들을 확인할수 있어야 한다. 여기서도 수자서명과 인증이 사용된다.
- **가장 좋은 보안실천들과 전자상거래에서 모든 합법적대방들을 보호하기 위한 체계설계기술들의 리용을 담보한다:** SET는 고도로 안전한 암호화알고리듬과 규약에 기초한 잘 검사된 명세서이다.
- **전송층의 보안기구에 의존하지 않고 또한 그러한 규약과 병용할수도 있는 규약을 준비할것:**SET는 "미가공"**TCP/IP**모임에서 안전하게 운영할수 있다. 그러나 SET는 IPSec와 SSL/TLS와 같은 다른 보안기구의 사용에 저촉되지 않는다.
- **쏘프트웨어와 망조달자들속에서 호상운영성을 장려하고 촉진한다:** SET규약과 형식들은 장치가동환경, 조작체계와 Web쏘프트웨어와 무관계하다.

SET의 기본특징

우에서 서술한 요구를 만족시키기 위해 SET는 다음과 같은 기능들을 가진다.

- **정보의 기밀성:** 카드소유자구좌와 지불정보는 그것이 망을 통하여 전송될 때 담보된다. SET의 흥미 있고 중요한 특징은 판매점이 카드소유자의 신용카드번호를 알수 없게 하는것이다. 즉 이것은 발행한 은행에만 제공된다. DES는 기밀성을 보장하는데 리용된다.
- **자료의 완정성:** 카드소유자로부터 판매점에게 전송되는 지불정보는 주문정보, 개인자료와 지불지시들을 포함한다. SET는 이 통보문내용들이 전송중에 변경되지 않도록 담보한다. RSA수자서명은 SHA-1하쉬코드를 리용하여 통보문완정성을 제공한다. 확실한 통보문들은 SHA-1를 사용한 HMAC에 의하여서도 보호된다.
- **카드소유자구좌인증:** SET는 카드소유자가 해당 카드구좌번호의 정당한 사용자라는것을 판매점이 확인할수 있게 한다. SET는 이를 위하여 RSA서명을 가지는 X.509v3수자증명서들을 리용한다.
- **판매점인증:** SET는 판매점이 재정기관과 련계를 가짐으로써 지불카드들을 접수할수 있다는것을 카드소유자가 확인할수 있게 한다. SET는 이를 위하여 RSA서명을 가진 X.509v3수자증명서를 리용한다.

SET는 IPSec와 SSL/TLS와는 달리 매 암호화알고리듬에 대하여 하나의 선택만을 제공한다. 이것은 SET가 단일요구모임을 가지는 단일응용이지만 IPSec와 SSL/TLS는 일정한 응용범위를 지원하도록 한것이기때문이다.

SET참가자

그림 14-8에 SET체계의 참가자들을 보여 주었다.

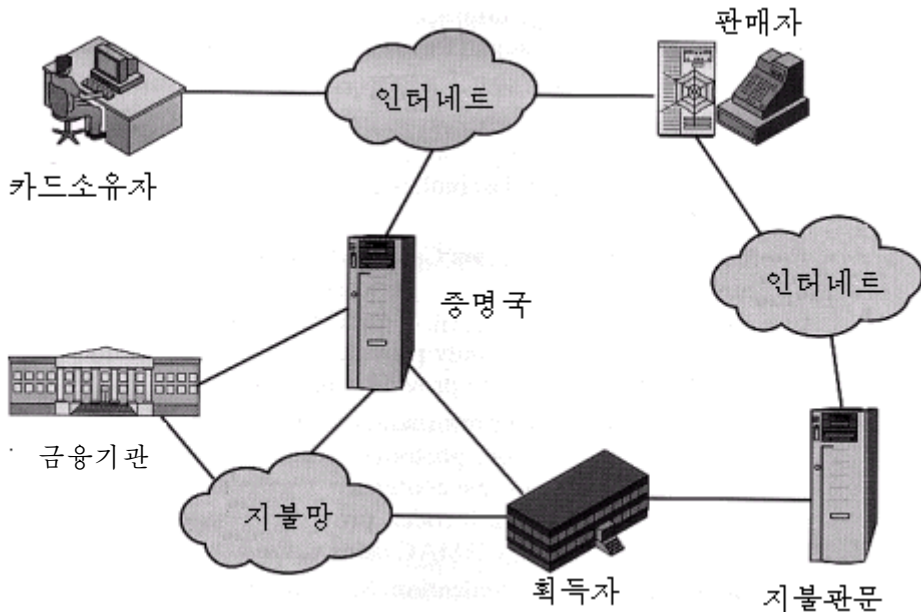



그림 14-8. 안전한 전자상업의 구성

- **카드소유자(cardholder):** 전자환경에서 소비자들과 회사구매자들은 인터네트상에서 개인용콤퓨터로 판매점들과 거래한다. 카드소유자는 발행자에 의하여 발행된 지불카드(실례로 MasterCard, Visa)의 공인된 소유자이다.
- **판매점(merchant):** 판매점은 카드소유자에게 상품들이나 봉사들을 팔아야 하는 개인 혹은 집단이다. 일반적으로 이 상품들과 봉사들은 Web싸이트를 거쳐 혹은 전자우편에 의하여 제공된다. 지불카드들을 접수한 판매점은 구매자와 련계를 가져야 한다.
- **금융기업(issuer):** 이것은 은행과 같은 금융기업들로서 카드소유자들에게 지불카드를 제공한다. 일반적으로 구좌들이 적용되며 우편으로 또는 직접 신청이 작성된다. 결국 카드소유자의 채무에 따르는 지불을 책임지는 발행이다.
- **획득자(acquirer):** 이것은 판매점와 함께 구좌를 확립하고 지불카드인증과 지불을 처리하는 금융기관들이다. 판매점들은 보통 한개 이상의 신용카드를 접수하는데 여러개의 신용대장이나 여러가지 개별적발행자들을 취급하려고 하지 않는다. 획득자는 주어 진 카드가 능동구좌이고 목적한 구매가 신용한계를 넘지 않는다는것을 판매점에 확인시킨다. 획득자는 또한 판매점의 구좌에 대한 지불의 전자전송도 제공한다. 다음으로 발행자는 획득자에게 어떤 지불망을 통하여 전자자금을 상환한다.

- **지불관문(payment gateway):**이것은 판매점지불통보문을 처리하는 획득자나 지정된 제3자에 의해서 조작되는 기능이다. 지불관문은 SET와 지불인증과 지불기능을 위한 기존지불카드지불망사이를 대면시킨다. 판매점들은 인터네트상에서 지불관문과 SET통보문을 교환하지만 그 지불관문은 획득자의 금융처리체계에 직접 혹은 망접속된다.
- **증명국(CA):**이것은 카드소유자, 판매점들 그리고 지불관문들에 X.509v3공개열쇠증명서들을 발행하는 신용되는 기관이다. SET의 성공은 이를 위하여 리용가능한 CA하부구조의 존재에 의존한다. 앞의 장들에서 론의한것처럼 CA들의 계층도가 리용되므로 참가자들이 기본증명국에 의하여 직접 증명되지 않아도 된다.

결재를 할 때 요구되는 사건렬을 보자. 이때 다음과 같은 상세한 암호학적과정을 고찰할수 있다.

1. **고객은 구좌를 연다.** 고객은 전자지불과 SET를 지원하는 은행과 MasterCard나 Visa와 같은 신용카드의 계약을 맺는다.
2. **고객은 증명서를 받는다.** 적당한 신원확인후 고객은 X.509v3수자증명서를 받는데 여기에는 은행이 서명한다. 증명서에는 고객의 RSA공개열쇠와 그것의 유효기간을 확인되여 있다. 그것은 또한 은행에 의하여 담보되는 고객의 열쇠쌍과 그의 신용카드사이의 관계를 확립한다.
3. **판매점은 매개의 증명서를 가진다.** 어떤 상표의 카드를 접수한 판매점은 그자신이 소유한 두개의 공개열쇠들에 대한 두개의 증명서(하나는 통보문에 서명하기 위한것이고 또 하나는 열쇠교환을 위한것)를 가지고 있어야 한다. 판매점은 또한 지불관문의 공개열쇠검증의 복사를 필요로 한다.
4. **고객이 주문한다.** 이것은 매개 판매점의 Web싸이트를 열람하여 항목들을 선택하고 가격을 정하는것을 포함하는 처리이다. 다음 고객은 판매점에 사려는 품종, 그 매개의 가격들, 총 가격, 주문번호목록을 포함하는 주문양식을 보낸다.
5. **판매점이 검증된다.** 주문양식외에 판매점은 자기 증명서의 복사를 보내고 결과 고객은 정당한 판매점을 대상하고 있다는것을 확인할수 있다.
6. **주문정보와 지불정보를 보낸다.** 고객은 판매점에 자기의 증명서와 함께 지불정보를 보낸다. 주문정보는 주문양식에서 매 항목들을 재확인하는데 쓰인다. 지불은 신용카드세부들을 포함한다. 지불정보는 판매점이 읽을수 없게 암호화된다. 고객의 증명서는 판매점이 고객을 확인할수 있게 한다.
7. **판매점은 지불권한을 요구한다.** 판매점은 고객이 가지고 있는 신용이 이 구매에 충분하다는것을 확인할것을 요구하면서 지불관문에 지불정보를 보낸다.
8. **판매점은 주문을 확인한다.** 판매점은 고객에게 주문에 대한 비준을 보낸다.
9. **판매점은 소비품이나 봉사들을 제공한다.** 판매점은 상품을 보내거나 고객에게 봉사를 제공한다.
10. **판매점은 지불을 요구한다.** 이 요구는 지불관문에로 전송되는데 그것은 모든 지불처리를 조종한다.

쌍대서명

SET규약의 세부를 고찰하기전에 SET에 도입된 중요한 성과인 쌍대서명을 론의하자. 쌍대서명의 목적은 서로 다른 두 수신자에게 보내려는 두개의 통보문을 련결하는것

이다. 이 경우에 고객은 주문정보(OI)를 판매점에 보내고 지불정보를 은행에 보낸다고 하자. 판매점은 고객의 신용카드번호를 알 필요가 없으며 은행은 고객의 주문에 대한 세부를 알 필요가 없다. 고객은 이 두 항목을 제각기 보관하며 개인성을 위하여 여가의 보호를 제공 받는다. 그러나 두 항목은 필요에 따라 론박을 해결하는데 리용할수 있도록 련결되여야 한다. 그것은 고객이 지불 대 주문관계를 밝히는데 필요하다.

련결에 대한 필요성을 밝히기 위해 고객이 두 통보문 즉 수표한 주문정보와 수표한 지불정보를 판매점에 보내고 판매점은 PI(지불정보)를 은행에 통과시킨다고 하자. 만일 판매점이 이 고객에게서 온 또 다른 주문정보를 받으면 판매점은 이 주문정보가 본래의 주문정보보다 더 좋은 지불정보를 준다고 생각할수 있다. 련결은 이것을 막는다.

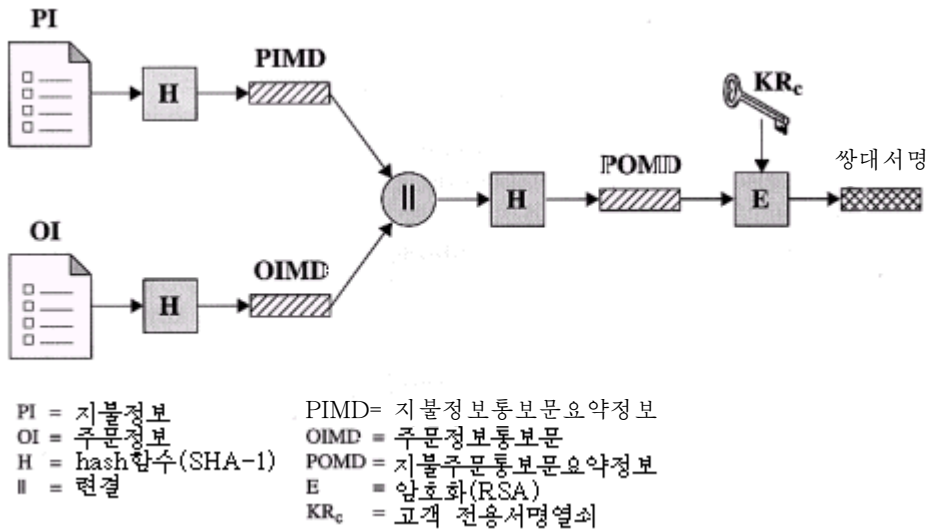



그림 14-9. 쌍대서명의 작성

그림 14-9에 앞에서 설명한 요구를 만족시키는 쌍대서명을 보여 주었다. 고객은 PI의 하쉬와 OI의 하쉬를(SHA-1를 리용하여) 취한다. 다음 이 두 하쉬들은 련결되고 그 결과의 하쉬가 취해 진다. 결국 고객은 자기의 비밀서명열쇠를 가지고 마지막하쉬를 암호화하고 쌍대서명을 창조한다. 조작은 다음과 같이 요약할수 있다.

$$DS=E_{KR_C}[H(H(PI)\|H(OI))]$$

여기서 KR_C는 고객의 비밀서명열쇠이다. 이때 판매점이 쌍대서명(DS)과 OI, 그리고 PI(PIMD)에 대한 통보문요약을 가지고 있다고 가정하자. 판매점도 역시 고객의 증명서로부터 얻어 진 그의 공개열쇠를 가지고 있다. 이때 판매점은 다음의 두개의 값을 계산할수 있다.

$$H(PIMD\|H(OI)) \text{ 및 } D_{KU_C}[DS]$$

여기서 KUc는 고객의 공개서명열쇠이다. 만일 이 두 량이 같다면 그때 판매점은 서명을 확인한것으로 된다. 마찬가지로 은행이 DS, PI, OI(OIMD)에 대한 통보문요약정보, 그리고 고객의 공개열쇠를 가지고 있으면 다음의것을 계산할수 있다.

$$H(H(PI)\|OIMD) \text{와 } D_{KU_C}[DS]$$

또한 이 두 량이 같으면 은행은 서명을 확인한다. 총체적으로

1. 판매점은 OI를 접수하고 서명을 확인한다.
2. 은행은 PI를 접수하고 그 서명을 확인한다.
3. 고객은 OI와 PI를 련결하고 그 결합을 검증할수 있다.

실례로 판매점이 거래에서 자기에게 유리하도록 다른 OI를 바꾸어 놓으려고 한다고 하자. 그러면 하쉬가 현재의 OIMD와 일치하는 다른 OI를 찾아야 한다. SHA-1에 대하여 이것은 실현할수 없을것이다. 결과 판매점은 이 PI와 다른 OI를 련결할수 없다.

지불처리

표 14-3에 SET에 의하여 지원되는 거래형태들을 보여 주었다. 다음의 거래들에서 그 일부를 상세히 보자.

- 구입요구
- 지불인증
- 매매처리

구입요구

구입요구(Purchase Request)교환에 앞서 카드소유자는 상품이나 봉사에 대한 열람, 선택 및 주문을 완료한다. 이 준비적인 단계는 판매점이 완료된 주문방식을 고객에게 보낼 때 끝난다. 이와 같은 선행한 모든것들은 SET를 리용하지 않고 진행된다.

구입요구교환은 4개의 통보문 즉 초기요구, 초기응답, 구입요구와 구입응답으로 구성된다. 판매점에게 SET통보문을 보내기 위하여 카드소유자는 판매점의 증명서의 복사와 지불관문을 가져야 한다. 고객은 판매점에 보낸 **요구개시통보문**에서 검증을 요구한다. 이 통보문은 고객이 리용하고 있는 신용카드의 상표를 포함한다. 통보문은 또한 고객에 의하여 이 요구/응답쌍으로 제정된 ID와 시기성을 담보하는데 리용되는 한번쓰기정보도 포함한다.

판매점은 응답을 하면서 자기 비밀서명열쇠로 거기에 서명한다. 응답은 고객으로부터의 한번쓰기정보, 고객이 통보문에 돌려 보낼 다른 한번쓰기정보 그리고 구입거래에 대한 거래ID를 포함한다. 서명된 응답외에 **응답개시통보문**에는 판매점의 서명증명서와 지불관문의 열쇠교환용증명서가 포함된다.

카드소유자는 판매점의 관문증명서들을 그들 각자의 CA서명들로 검증하고 다음OI와 PI를 작성한다. 판매점에 의하여 할당된 거래ID는 OI와 PI들에 다 배치된다. OI는 상품들의 값과 수량과 같은 명백한 주문자료를 포함하지 않는다. 그대신 그것은 첫 SET통보문전의 물건을 사는 단계에서 판매점과 고객사이의 거래과정에 생성되는 주문참조정보를 포함한다. 다음 카드소유자는 **구입요구**통보문을 준비한다(그림 14-10). 이를 위하여 카드소유자는 1회용대칭암호열쇠 Ks를 생성한다. 통보문은 다음과 같은 정보를 포함한다.

표 14-3. SET거래형

카드소유자등록	카드소유자들은 자기들이 SET통보문을 판매점에게 보내기전에 CA에 등록하여야 한다.
판매점등록	판매점은 그들이 고객과 지불관문을 포함한 SET통보문을 교환하기전에 CA에 등록하여야 한다.
구입요구	고객으로부터 판매점에 보내는 판매점의 OI와 은행의 PI를 포함한 통보문.
매매처리	신용카드계산서에 기초한 구입에 대하여 주어 진 량을 담보하는 판매점과 지불관문사이의 교환.
지불인증	판매점과 지불관문사이의 통보문교환에서 어떤 지불이 그 고객의 신용카드구좌에서 지불가능한가 하는것을 조사한다.
증명서들사이의 대면	CA가 검증요구의 처리를 빨리 완료할수 없으면 그것은 요구자가 후에 검사하여 보낼것을 표시하면서 판매점과 카드소유자에게 응답을 보낼것이다. 카드소유자나 판매점은 검증요구의 상태를 결정하고 요구가 승인되면 검증을 접수하는 검증조사통보문을 보낸다.
구입조사	카드소유자가 구입응답을 접수한후에 주문처리상태를 검사할수 있다. 이 통보문은 인증, 매매, 신용처리 등의 상태를 표시하지만 다시 주문된 상품의 상태와 같은 정보를 포함하지 않는다는것을 주의하시오.
인증취소요구	판매점이 이전의 지불인증을 바로 잡도록 한다. 주문이 완료되지 않으면 판매점은 지불인증자체를 취소한다. 주문의 일부가 완료되지 않으면 판매점은 지불인증의 일부만을 취소한다.
매매취소요구	판매점에 의하여 정확치 않게 입력된 거래활동과 같은 구입요구에서 판매점이 오유를 바로 잡도록 한다.
환금처리	판매점이 상품이 되돌아 왔을 때와 파는 동안 파손될 때 카드소유자의 계산서에 신용을 발행하도록 한다. SET신용통보문은 항상 카드소유자가 아니라 판매점에 의하여 초기화된다는것을 주의하시오. 처리된 신용에 기인하는 카드소유자와 판매점사이의 통신은 SET의 밖에서 일어 난다.
환금취소	판매점이 이미 진행한 환금처리를 바로 잡도록 한다.
지불관문증명서요구	판매점이 지불관문을 조사하고 관문의 현재열쇠와 서명검증의 복사를 접수하도록 한다.
묶음관리	판매점이 지불관문과 판매점의 묶음에 대한 정보를 취급하는데 쓰인다.
오유통보문	응답자가 그것이 형식화 혹은 내용확인검사를 하지 못하기때문에 통보문을 접수하지 못하는 경우에 리용한다.

1. **구입관련정보.** 이 정보는 판매점에 의하여 지불관문에 부쳐 지는데
   - ¤ PI
   - ¤ PI와 DI우에서 처리되고 고객의 전용서명열쇠로 수표된 쌍대서명
   - ¤ OI통보문요약정보(OIMD)

   로 이루어 진다.

   OIMD는 앞에서 설명한것처럼 지불관문과 쌍대서명을 확인하는데 필요하다. 이 모든 항목들은 Ks로 암호화된다. 마지막항목은

   - ¤ 전자봉투이다. 이것은 지불관문의 공개열쇠교환열쇠로 Ks를 암호화하여 만들어 진다. 이전에 기록된 다른 항목들을 읽기전에 봉투가 열려야(복호되여야) 하므로 그것을 전자봉투라고 한다.

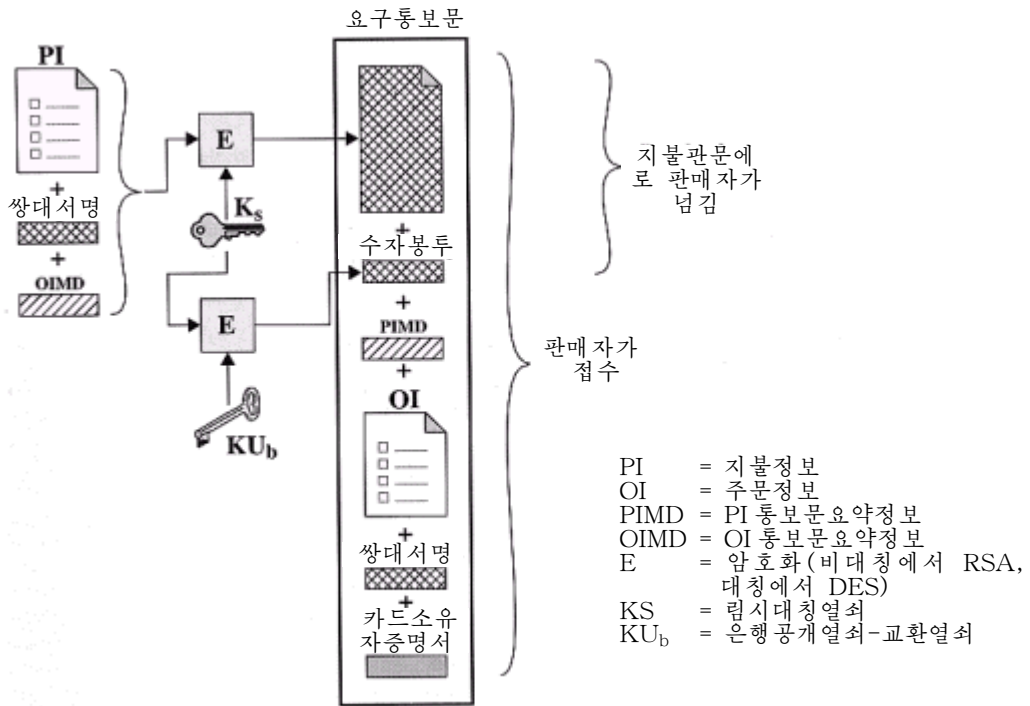



그림 14-10. 카드소유자가 보내는 구입요구통보문

Ks의 값은 판매점에게 리용가능하게 만들어 지지 않는다. 따라서 판매점은 어떠한 지불관련정보도 읽을수 없다.

2. **주문관련정보** .이 정보는 판매점에 의하여 요구되며
   ¤ OI
   ¤ PI와 OI에서 처리되고 고객의 비밀서명열쇠로 수표된 쌍대서명
   ¤ PI통보문요약정보(PIMD)

   로 이루어 졌다.

   PIMD는 판매점이 쌍대서명을 확인하는데 필요하다. OI는 명백히 보내진다는것을 주의하시오.
3. **카드소유자증명서**.이것은 카드소유자의 공개서명열쇠를 포함한다. 그것을 판매점과 지불관문이 요구한다.

판매점이 구입요구통보문을 접수하면 그것은 다음의 동작(그림 14-11)을 수행한다.

1. 자기의 CA서명을 리용하여 카드소유자의 증명서를 검증한다.
2. 고객의 공개서명열쇠를 리용하여 쌍대서명을 검증한다. 이것은 주문이 전송도중에 변경되지 않았으며 그것이 카드소유자의 전용서명열쇠로 서명되였다는것을 담보한다.
3. 주문을 처리하고 지불정보를 인증을 위하여 지불관문에 보낸다(후에 설명).

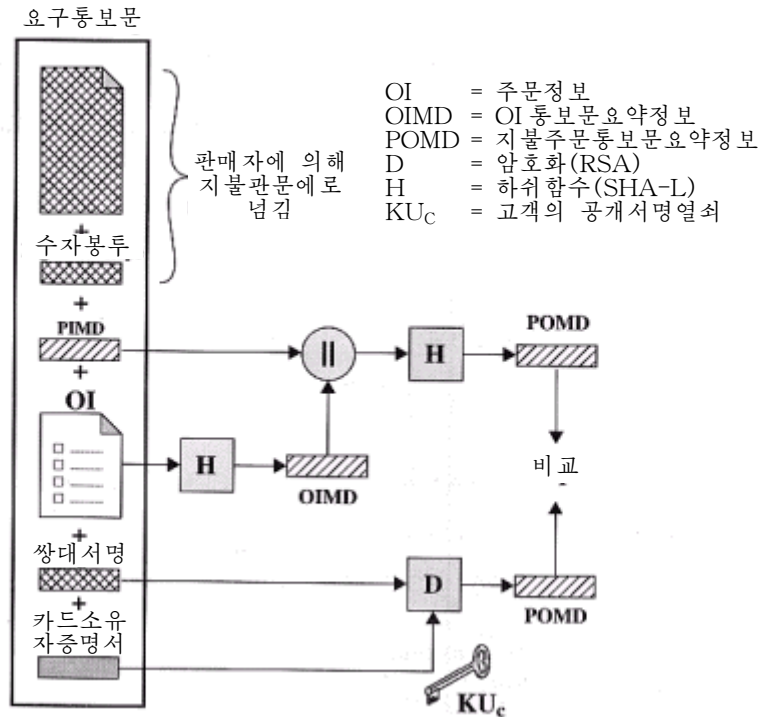



그림 14-11. 판매자가 고객의 구매요구를 확인

4. 고객에게 구입응답을 보낸다.

구입응답통보문은 주문을 알고 해당한 거래번호를 참조하는 응답블로크를 포함한다. 이 블로크는 판매점에 의하여 판매점의 전용서명열쇠로써 서명된다. 블로크와 그 서명은 판매점의 서명검증용증명서와 함께 고객에게 보내진다.

카드소유자의 쏘프트웨어는 구입응답통보문을 접수할 때 먼저 판매점의 증명서를 확인하고 다음 응답블로크에 대한 서명을 확인한다. 마지막으로 그것은 주문의 상태에 의하여 자료기지를 갱신하거나 사용자에게 통보문을 현시하는것과 같은 응답에 기초한 동작을 한다.

지불인증

카드소유자로부터의 주문을 처리할 때 판매점은 그 거래에 대하여 지불관문으로 지불인증한다. 지불인증은 거래가 금융기관에 의하여 승인된다는것을 담보한다. 권한은 판매점이 지불을 접수할것을 담보한다. 즉 판매점이 고객에게 봉사나 상품을 제공할수 있게 한다. 지불인증교환은 두 통보문 즉 인증요구와 인증응답으로 구성된다.

판매점은 다음의것으로 이루어 지는 지불에 **인증요구**통보문을 보낸다.

1. **구입관련정보:** 이것은 고객으로부터 얻어 지며 다음의것들이 포함된다.
   ¤ PI
   ¤ PI와 OI에서 계산되고 고객의 비밀서명열쇠로 서명된 쌍대서명
   ¤ OI통보문요약정보(OIMD)
   ¤ 전자봉투
2. **인증관련의 정보:** 이 정보는 판매점에 의하여 생성되는데
   ¤ 판매점의 전용서명열쇠로 서명된 거래를 포함하는 권한블로크와 판매점에 의하여 생성된 1회용대칭열쇠로 암호화된 거래ID를 포함하는 권한블로크
   ¤ 전자봉투. 이것은 지불관문의 열쇠교환용공개열쇠로 암호화하여 작성된다.
3. **증명서:** 판매점은 카드소유자의 서명열쇠증명서(쌍대서명를 확인하는데 리용), 판매점의 서명열쇠증명서(판매점의 서명을 확인하는데 리용),판매점의 열쇠교환 증명서(지불관문의 응답에 필요)들을 소유한다.

지불관문은 다음의 작업을 진행한다.

1. 모든 증명서들을 확인한다.
2. 인증정보블로크의 전자봉투를 복호하여 림시열쇠를 도출하고 인증정보블로크를 복호한다.
3. 인증블로크에 있는 판매점의 서명을 검증한다.
4. 대칭열쇠를 얻기 위한 인증블로크의 전자봉투를 복호하여 림시열쇠를 도출한 다음 지불블로크를 복호한다.
5. 지불블로크에 있는 쌍대서명을 확인
6. 판매점으로부터 접수한 거래ID가 고객으로부터(간접적으로) 접수한 PI에서의것과 일치한다는것을 검증한다.
7. 금융기관으로부터 인증을 요구하고 접수한다.

금융기관으로부터 인증정보를 얻으면 지불관문은 판매점에 **인증응답**통보문을 되돌린다. 그것은 다음과 같은 요소를 포함한다.

1. **권한관련정보:** 관문의 전용서명열쇠로 수표된 인증블로크를 포함하고 관문에 의하여 생성된 대칭열쇠로 암호화한다. 또한 판매점들의 1회용대칭공개열쇠교환용 열쇠로 암호화한 수자식봉투를 포함한다.
2. **획득통표정보:** 이 정보는 후에 효과적인 지불에 리용된다. 이 블로크는 전자봉투와 함께 서명되고 암호화된 획득통표와 같은 형태이다. 이 통표는 판매점에 의하여 처리되지 않는다. 그것은 지불요구와 함께 귀환되여야 한다.
3. **증명서:** 관문의 서명열쇠증명서

관문으로부터의 인증을 접수하고 판매점은 고객에게 상품과 봉사를 제공한다.

매매처리

판매점은 지불관문과의 사이에서 매매요구통보문과 매매응답통보문에 의한 매매처리를 진행하고 지불을 받는다.

판매점은 **포착요구**(capture request)를 위하여 포착요구블로크를 서명하고 암호화하는데 그것은 지불량과 거래ID를 포함한다. 통보문은 역시 판매점의 서명열쇠나 열쇠교환열쇠증명서와 마찬가지로 이 매매를 위하여 이미(인증응답에서) 접수된 암호화된 획득증거를 포함한다.

지불관문이 매매요구통보문을 수신하면 그것은 매매요구블로크를 복호하고 검증하며 획득증거블로크를 복호하고 검증한다. 다음 그것은 매매요구와 매매증거사이의 일치성에 대한 검사를 한다. 그리고 그것은 개별지불망에서 금융기관에 보낸 청구서를 참조한다. 그러면 판매점의 구좌에 지불이 진행된다.

다음 관문은 지불이 되였다는것을 판매점에 **포착응답**통보문으로 알려 준다. 이 통보문에는 관문이 서명하고 암호화한 매매응답블로크가 포함된다. 또한 통보문은 관문의 서명용열쇠증명서도 포함한다. 판매점의 쏘프트웨어는 후에 지불정보를 참회하기 위하여 매매응답을 보관해 둔다.

참고문헌

[GARF97] Garfinkel, S., and Spafford, G. *Web Security & Commerce*. Cambridge, MA: O'Reilly and Associates, 1997.,

[MACG97] Macgregor, R.; Ezvan, C.; Liguori, L.; and Han, J. *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*. IBM RedBook SG24-4978-00, 1997. Available at www.redbooks.ibm.com/SG244978.

[RUBI97] Rubin, A.; Geer, D.; and Ranum, M. Web *Security Sourcebook*. New York: Wiley, 1997.

참고할 Web싸이트들

- **Netscape's SSL Page**: SSL 명세서를 포함한다.
- **Transport Layer Security Charter**: 최근의 RFC들과 TLS용호상련결망초안.
- **MasterCard SET Site**: 최근의 SET문서들과 부록들 그리고 응용정보들.
- **Visa-Electronic Commerce Site**: MasterCard Site에서와 같은 정보.

문 제

1. SSL과 TLS에서 왜 change_cipher_spec통보문이 핸드쉐이크규약에 포함되지 않고 개별적으로 존재하는가?
2. Web보안에 대한 다음의 위협들을 고찰하고 그 매개가 SSL의 기능에 의해 어떻게 방지되는가를 설명하시오.
   ㄱ) 힘내기공격: 전통암호알고리듬에서 열쇠공간에 대한 전수탐색
   ㄴ) 기지평문사전공격: 많은 통보문들의 HTTP GET지령과 같은 예상할수 있는 평문을 포함한다. 공격자는 기지평문통보문의 모든 가능한 암호화를 포함하는 사전을 구성한다. 암호화된 통보문이 도청되면 공격자는 암호화된 기지평문을 포함하는 부분을 취하여 사전에서 암호문을 찾는다. 암호문은 같은 비밀열쇠로 암호화된 입력과 일치하여야 한다. 만일 일치하는것이 여러개 있으면 적당한것을 결정하기 위해 충분한 암호문에 대하여 그 매개를 시험해 볼수 있다. 이 공격은 특히 크기가 작은 열쇠들에 대하여 효과적이다(실례로 40-bit열쇠).
   ㄷ) 재연공격: SSL핸드쉐이크통보문들이 대면된다.
   ㄹ) 중간대조공격: 공격자가 봉사기에 대하여 의뢰기로 행동하고 의뢰기에 대해서는 봉사기처럼 행동하면서 열쇠교환과정에 끼여 들수 있다.
   ㅁ) 통과암호의 예측: HTTP 또는 다른 응용거래들에서 통과암호들이 도청된다.
   ㅂ) IP기만: 위조자료를 접수하도록 위조된 IP주소를 리용한다.
   ㅅ) IP강탈: 두 가입자들사이의 능동적이며 인증된 접속을 호환시켜 공격자는 그 가입자들중의 하나로 참가한다.
   ㅇ) SYN범람: 공격자는 접속을 요구하는 TCP SYN통보문을 보내는데 그 옹근 접속을 확립하는 마지막통보문과는 응답하지 않는다. 공격 받은 TCP모듈은 일반적으로 몇분동안 《절반 열린 접속》을 남긴다. 반복된 SYN통보문은 TCP모듈을 방해할수 있다.
3. 이 장에서 배운것에 기초하면 SSL에서 수신자가 정상적으로 도착한 SSL레코드블로크들을 다시 주문할수 있는가? 만일 그렇다면 어떻게 할수 있는가를 설명하시오. 그렇게 할수 없다면 왜 그런가?

제4편. 체계보안

제15장. 침입자, 비루스와 웜

망체계에서 중요한 보안문제는 사용자들이나 쏘프트웨어에 의한 적의를 품은 침해 혹은 적어도 원하지 않는 **침해**이다. 사용자침해는 권한이 부여되지 않은 콤퓨터에 가입하는 형태 혹은 권한을 가진 사용자인 경우에 자기의 권한을 벗어 나서 특권을 획득하거나 행동을 진행하는 형태로 나타날수 있다. **쏘프트웨어침해**는 비루스,웜 혹은 트로이목마의 형태를 취할수 있다.

이 모든 공격들은 망을 통해서 체계에 가입할수 있으므로 망보안과 관련된다. 그러나 이 공격들은 망에 기초한 공격에만 머무르는것이 아니다. 국부말단까지 접근한 사용자는 중간망을 리용하지 않고 침해하려고 할수 있다. 비루스나 트로이목마는 디스크에 의해 체계에 들어 갈수 있다. 웜만은 독특한 망현상이다. 이렇게 체계침해문제는 망보안과 **콤퓨터보안**전반에 관계된다.

이 책에서는 망보안이 기본이므로 공격이나 체계침해와 관련한 보안대응수단을 종합적으로 해석하지 않는다. 그대신 이 장에서는 이 문제들에 대해 폭넓은 개괄을 준다.

이 장에서는 먼저 침입에 대해서 고찰한다. 우선 공격의 본질,예방전략과 방어가 실패하는 경우 침입을 검출하기 위한 전략에 대해서 고찰한다. 다음으로 널리 알려 진 비루스문제를 고찰한다.

15.1 침입자

가장 널리 알려 진 보안에 대한 두가지 위협가운데서 하나는 침입자인데(다른 하나는 비루스이다.) 일반적으로 해커나 크랙커를 의미한다. 자기의 중요한 첫 연구에서 앤더슨(Anderson)이 세가지 부류의 침입자들을 밝혀 냈다[ANDE80].

- **가장자:**콤퓨터를 리용할 권한은 없이 합법적인 사용자의 등록자리를 부당하게 리용하려고 체계의 접근조종에 침투하는 사람이다.
- **불법행위자:**접근할 권한이 없는 자료,프로그람,자원에 접근하거나 이런 접근에 대한 권한은 있지만 권한을 나쁜 목적으로 리용하는 합법적인 사용자이다.
- **도용자:**체계의 감시조종을 잘 알고 검열과 접근조종을 피하거나 검열수집을 억제하기 위해 감시조종을 리용하는 개별적인 사람이다.

가장자는 외부침입자이고 불법행위자는 일반적으로 내부침입자이며 도용자는 외부침입자이든가 내부침입자일수 있다.

침입자의 공격범위는 가벼운 공격으로부터 엄중한 공격에 이르기까지 넓은 범위이

다. 가벼운 공격은 단순히 인터네트를 조사하고 거기에서 무엇인가를 찾아 보려고 하는것이고 엄중한 공격은 비밀자료를 읽고 거기에 비법적인 변경을 가하거나 체계를 파괴하려는것이다.

침입자의 위협은 특히 클리프 스톨(**Cliff Stoll**)이 기록한 유명한 《음흉한 해커》(《Wily Hacker》)사건(1986-1987년)으로 해서 널리 알려 지게 되였다[STOL88, STOL89]. 1990년에 콤퓨터해커들에 대한 전반적인 단속이 있었다[STER92]. 그후 많은 사람들이 이 문제는 이제는 해결되였으리라고 믿고 있었다.

그러나 사실 이 문제는 해결되지 못하고 있다. 한가지 실례로서 벨 라브(Bell Lab) [BELL92, BELL93]의 그룹은 인터네트를 통해서 자기들의 콤퓨터복합체가 오랜 기간 각이한 곳으로부터 자주 공격을 받았다고 발표하였다. 그당시 벨(Bell)그룹에서는 다음과 같은 일들이 벌어 지고 있었다.

- 매일 한번이상의 정도로 통과암호파일을 복사하려고 한다.
- 의심스러운 원격수속호출(RPC)이 하루에 한번이상의 정도로 있다.
- 적어도 두 주에 한번씩 존재하지 않는 《미끼》콤퓨터에 접촉하려고 한다.

가벼운 침입자들에 대해서는 비록 그들이 자원을 소비하고 합법적인 사용자들의 운영에 지장을 줄지라도 좀 참을수 있다. 문제는 침입이 가벼운것인가, 매우 엄중한것인가를 미리 아는 방도가 없다는것이다. 따라서 특별히 비밀이 아닌 자원을 가지고 있는 체계일지라도 이 문제를 취급해야 한다.

텍사스 A&M종합대학에서 일어 난 위협이 그 단적인 실례이다[SAFF93]. 1992년 8월에 이 대학에 있는 콤퓨터쎈터에 그곳 콤퓨터들중 한대가 인터네트를 통해 다른 지역에 있는 콤퓨터를 공격하는데 리용되였다고 알려 왔다. 조사결과 콤퓨터쎈터의 연구사들은 각이한 외부침입자들이 있다는것을 알게 되였다. 그 침입자들은 각이한 콤퓨터상에서 통과암호-크랙킹루틴들을 실행하고 있었다(싸이트에는 모두 12000대의 콤퓨터들이 접속되여 있었다). 쎈터는 영향을 받은 콤퓨터들을 분리하여 로출된 보안구멍들을 막은 다음 다시 정상조작을 시작하였다. 며칠후에 국부체계의 어느 한 관리자는 침입자의 공격이 다시 시작되였다는것을 검출하였다. 그 공격은 생각했던것보다 훨씬 더 교묘하였다. 관리자는 안전하다고 생각하는 일부 주요 봉사기들에서 발견된 수백개의 로출된 통과암호들을 가진 파일들을 발견하였는데 그중에는 안전하리라고 보았던 주봉사기안의 일부 파일들도 있었다. 게다가 한대의 국부콤퓨터는 해커들이 서로 접촉하여 기술과 개발과정을 론의하군 하는 해커용의 게시판으로까지 리용되고 있었다.

이 공격에 대한 분석으로부터 실제로는 두개의 해커준위가 존재한다는것이 밝혀 졌다. 높은 준위는 기술을 완전히 정통한 교묘한 사용자였고 낮은 준위는 크랙킹프로그람들이 어떻게 동작하는가는 거의나 모르면서 제공된 그대로 사용만 하군 하는《풋내기》였다. 이 협동동작은 침입자무기고에 있는 두가지 엄중한 무기 즉 침입방법에 대한 고급한 지식과 약점을 철저히 조사하기 위해 많은 시간을 바치려는 의지가 결합되게 하였다.

침입자문제의 중요성에 대한 각성이 높아 진 결과 **콤퓨터비상대책팀**(CERT)들이 창설되였다. 이 협력기관들은 체계의 약점에 대한 정보를 수집하고 그것을 체계관리자에게 알려 주었다. 공교롭게도 해커들은 CERT의 보고문을 얻을수 있었다. 텍사스 A&M사건에 대한 이후의 분석은 해커들이 CERT가 발표하였던 거의 모든 약점들에 대해서 바로 공격된 콤퓨터들을 시험해 볼수 있는 프로그람을 개발하였다는것을 보여 주었다. 한대의 콤퓨터라도 CERT의 권고에 따라 즉시 대책을 세우지 못했더라면 이와 같은 공격에 넓

은 길을 열어 주었을것이다.

통과암호크랙킹프로그람을 실행하는것외에 침입자들은 체계에 가입하고 있는 사용자들의 통과암호를 발견하기 위해 가입쏘프트웨어를 변경하려고 시도하였다. 이 과정에 해커들은 매우 위험한 통과암호들의 특징적인 모임을 만들수 있게 되였는데 그것을 피해자들의 콤퓨터들중 한 콤퓨터상에 설정된 게시판에서 리용할수 있게 되였다.

이 절에서는 먼저 침입기술에 대하여 고찰하고 다음 침입을 예방하는 방법들을 고찰한다. 예방에서 실패하면 방어의 두번째 선은 침입검출인데 이것은 마지막부분에서 론의한다.

침입기술

침입자의 목표는 체계에 접근할 자격을 얻거나 체계에 접근할수 있는 특권범위를 넓히는것이다. 그러자면 일반적으로 침입자가 보호되여야 할 정보를 입수해야 한다. 대부분의 경우에 이 정보는 사용자의 통과암호형태로 나타난다. 어떤 다른 통과암호를 알면 침입자는 그 체계에 가입할수 있으며 합법적인 사용자와 같이 모든 특권을 행사할수 있다.

일반적으로 체계에는 권한이 있는 매 사용자가 가지고 있는 통과암호와 관련한 파일을 보존되여 있다. 이런 파일이 보호없이 보관되여 있으면 해커들이 그것에 접근하여 통과암호를 쉽게 알수 있다. 통과암호파일은 다음의 두 방법가운데 한가지 방법으로 보호할수 있다.

- **한방향암호화:**체계는 사용자의 통과암호를 암호화된 형태로만 기억한다. 사용자가 통과암호를 제출하면 체계는 그것을 암호화하여 기억된 값과 비교한다. 실천에서 체계는 보통 암호화함수에 대한 열쇠를 생성하는데 통과암호가 리용되고 고정길이의 출력이 생성되는 한방향변환을 진행한다.
- **접근조종:**통과암호파일에로의 접근은 하나 혹은 몇개의 등록자리로 제한된다.

이것들중 한개 혹은 두개의 대응수단을 가지게 되면 침입자가 통과암호를 알아 내는데 일정한 품이 들게 된다. 많은 문헌들과 통과암호크랙커들과의 인터뷰자료에 대한 조사에 기초하여[ALVA90] 통과암호를 배우는 다음의 방법들을 소개한다.

체계에 들어 있는 표준등록자리로서 리용되는 음적통과암호를 조작해 보시오. 많은 관리자들은 이 음적통과암호를 변화시키려고 하지 않는다.

1. 짧은 통과암호를 모두 입력시켜 보시오(한개부터 세개의 문자까지).
2. 체계의 직결식사전이나 적절한 통과암호목록에 있는 단어를 입력시켜 보시오. 후자의 실례들은 해커게시판상에서 쉽게 볼수 있다.
3. 사용자들의 완전한 이름,그들의 안해와 아이들의 이름,사무실에 걸려 있는 그림들,그들의 취미와 관련되는 책 등과 같은 사용자들에 대한 정보를 수집해 보시오.
4. 사용자의 전화번호,사회적인 보안번호,방번호를 입력시켜 보시오.
5. 이 상태에 대한 모든 합법적인 허가번호를 입력시켜 보시오.
6. 15.2절에서 설명하는 접근에 대한 제한을 우회하는 트로이목마를 리용해 보시오.
7. 원격사용자와 가입자체계사이의 선을 도청해 보시오.

처음의 6가지 방법은 통과암호를 추측하기 위한 각이한 방법이다. 침입자가 가입을 시도하면서 검증해 보아야 한다면 그것은 지루하면서도 쉽게 부닥칠수 있는 공격방법이

다. 실례를 들어 체계는 통과암호가 세번 입력된 다음에는 그 어떤 가입도 거부하므로 가입자에 다시 접속하려는 침입자는 다시 한번 입력시켜 보아야 한다. 이러한 상태에서 통과암호를 더이상 입력시켜 보는것은 다문 몇개라도 통과암호를 가지는것보다 실천성이 없다. 그러므로 침입자가 이와 같은 방법을 그대로 리용하는것은 적합하지 않다. 례를 들어 침입자가 암호화된 통과암호파일에 낮은 특권준위로 접근할수 있다면 침입계획은 이 파일을 포착한 다음 더 높은 특권을 제공하는 옳은 통과암호를 발견할 때까지 천천히 이 체계의 개개의 암호기구를 리용하는것이다.

추측공격이 실현가능하고 많은 추측회수가 자동적으로 시도될수 있으며 추측공정자체가 검출됨이 없이 매 추측을 검증할수 있을 때 실지로 이것은 매우 효과적이다. 이 절의 마지막에 추측공격을 막는 방법에 대하여 본다.

앞에서 설명한 7번째 공격방법 즉 트로이목마는 막기가 특히 힘들수 있다. 우회접근조종실례프로그람은 [ALVA90]에 소개되여 있다. 낮은 준위 특권사용자는 오락프로그람을 만들고 체계조작자에게 그것을 여가시간에 리용하도록 권고한다. 프로그람은 실지 오락기능을 수행하지만 그 배경에는 역시 암호화되여 있지 않으나 접근은 보호되여 있는 통과암호파일을 사용자파일에 복사하는 부호가 포함되여 있다. 오락은 조작자의 높은 특권준위방식에서 실행되므로 통과암호파일에 대한 접근을 얻을수 있다.

8번째 공격 즉 선로도청은 물리적인 보안문제이다. 이때 5. 1절에서 론의된 련결암호화기술문제와 부닥칠수 있다.

이제는 두가지 중요한 대응수단 즉 예방과 검출의 론의에로 돌아 가자. 예방은 매력적인 보안목표이며 항상 힘들다. 방어자는 모든 가능한 공격을 막아야 하므로 힘들지만 공격자는 방어고리에서 가장 약한 고리를 찾아 이 점에 공격하므로 쉽다. 검출은 공격에 대하여 아는것이며 그것이 성공한 후이든가 전에 진행된다.

통과암호보호

침입자를 막는 방어의 전초선은 통과암호체계이다. 대부분 모든 다중사용자체계는 사용자의 이름이나 식별자(ID)뿐아니라 통과암호도 제공할것을 요구한다. 통과암호는 체계에 개별적으로 가입하는 ID를 인증하는데 봉사한다. ID는 다음의 방법으로 보안을 제공한다.

- ID는 사용자가 체계에 접근할수 있는 권한을 가지고 있는가를 결정한다. 일부 체계에서는 체계의 파일에 있는 ID를 가지고 있는 사람만이 접근을 할수 있게 한다.
- ID는 사용자에게 부여된 특권을 결정한다. 일부 사용자들은 감독이거나 파일을 읽을수 있는 《슈퍼사용자》이여야 하며 조작체계에 의해서 특별히 보호된 기능을 실행한다. 일부 체계들은 손님용 혹은 밝혀 지지 않은 등록자리를 가지고 있으며 이 등록자리의 사용자들은 다른 사용자들보다 더 제한된 특권을 가진다.
- ID는 자위적접근조종을 하는데 리용된다. 례를 들어 다른 사용자의 ID를 렬거하는것으로 사용자는 그 사용자가 소유하고 있는 파일들을 읽을수 있는 허락을 받을수 있다.

통과암호의 약점

공격의 본질을 리해하기 위해서 UNIX체계에서 널리 리용되는 도식을 생각해 보자. 이 체계에서 통과암호들은 절대로 명백한 형태로 보관되지 않는데 다음과 같은 절차를 쓰고 있다(그림 15-1의 ㄱ). 매 사용자는 기호의 길이가 8까지인 통과암호를 선택한다.

이것은 통파암호루틴의 열쇠입력으로 되는 56-bit값(7bit는 아스키(ASCII)부호를 리용한다.)으로 변환된다. 암호(3)의 암호루틴은 DES에 기초하고 있다. DES알고리듬은 12-bit《쏠트(salt)》값을 리용하여 변경된다. 일반적으로 이 값은 통파암호가 사용자에게 할당될 때의 시간과 관련된다. 변경된 DES알고리듬은 령인 64-bit블로크입력자료로 동작한다. 알고리듬의 출력은 다음 두번째 암호화의 입력자료로 된다. 이 처리공정은 25번의 암호화 전과정에 대하여 반복된다. 64-bit출구결과는 11개의 기호렬로 변환된다.

ㄱ) 새 통파암호를 넣기

통파암호파일

사용자 id

사용자 id 쏠트 E_{pwd}[쏠트, 0]

쏠트

선택

통파암호

암호(3)

암호화된 통파암호

비교

ㄴ) 통파암호의 검증

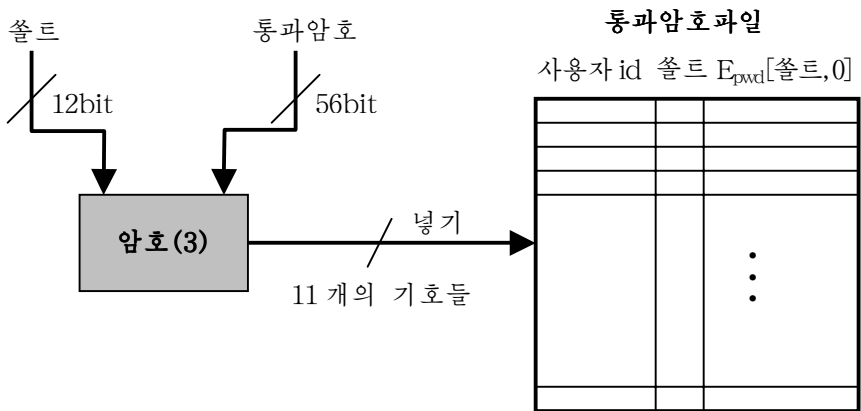
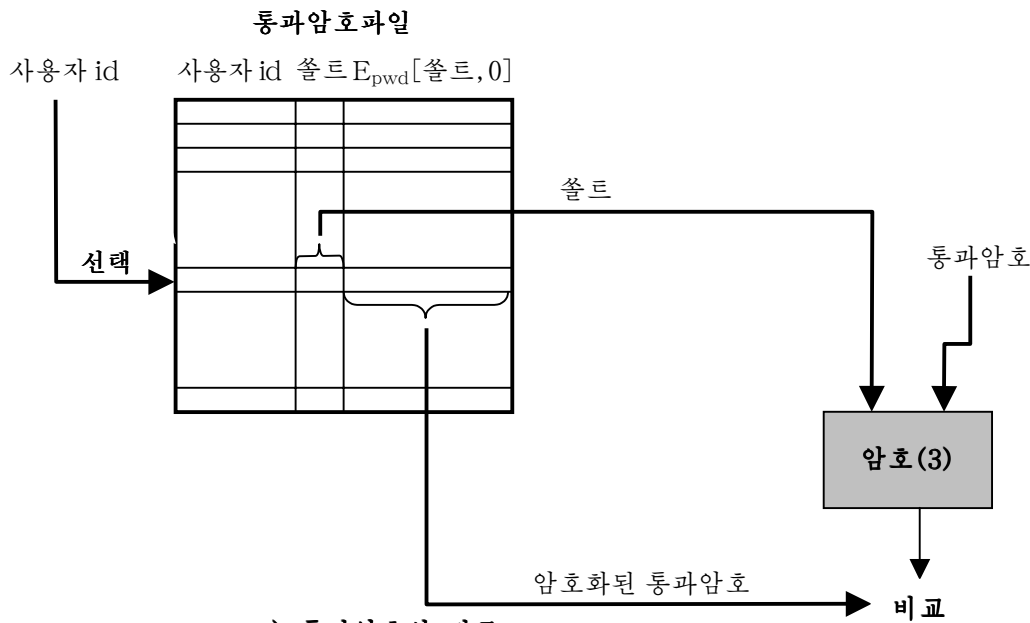



그림15-1. UNIX통파암호도식

통과암호의 암호문은 쏠트를 복사한 평문과 함께 사용자의 식별자에 대응하는 통과암호파일에 기억시킨다.

쏠트는 다음의 세가지 목적으로 쓰인다.

- 그것은 통과암호파일에 나타나는 통과암호들이 중복되지 않게 한다. 두 사용자가 같은 통과암호를 선택하였다 하더라도 이 통과암호들은 다른 시간에 할당되였을것이다. 그러므로 두 사용자의 《확장된》통과암호는 서로 다른것이다.
- 그것은 사용자가 두개의 추가적인 기호렬들을 기억할 필요가 없이도 통과암호의 길이를 효과적으로 증가시킨다. 그러므로 가능한 통과암호의 수는 4096의 인수로 늘어 나며 동시에 통과암호를 추측하는것도 더 어려워 진다.
- 그것은 힘든 란폭공격을 쉽게 할수 있는 DES의 하드웨어실현을 리용하지 못하게 한다.

사용자는 UNIX체계에 가입하려고 할 때 식별자와 통과암호를 제공한다. 조작체계는 사용자의 식별자를 리용하여 통과암호파일을 색인하며 암호화루틴의 입력으로 리용되는 평문쏠트와 암호화된 통과암호를 돌려 준다. 결과가 기억된 값과 같으면 통과암호는 접수된다.

암호루틴은 추측공격을 막을수 있도록 설계한다. DES로 작성된 쏘프트웨어는 하드웨어판본에 비해 느리며 25번의 반복리용은 요구되는 시간을 25배로 되게 한다. 그러나 이 알고리듬의 초기설계로부터 두가지 변화가 일어 난다. 첫째로, 알고리듬을 더 새롭게 실현한것은 속도를 더 빠르게 하는 결과를 가져 온다. 례를 들어 인터네트웜은 공격한 UNIX체계에 기억된 규격보다 더 효률적인 암호알고리듬을 리용하여 매우 짧은 시간에 수백개의 통과암호에 대한 직결식통과암호추측을 할수 있다. 둘째로, 하드웨어의 효률이 계속 높아 짐에 따라 임의의 쏘프트웨어알고리듬을 더 빨리 실행할수 있다.

따라서 UNIX의 통과암호도식에는 두가지 위협이 존재한다. 첫째로, 사용자는 림시적인 등록자리를 리용하거나 어떤 다른 수단으로써 콤퓨터에 대한 접근권한을 얻은 다음 그 콤퓨터에서 통과암호크랙커라고 부르는 통과암호를 추측하는 프로그람을 실행할수 있다. 공격자는 적은 자원을 소비하면서 수백 혹은 수천개의 가능한 통과암호를 검사하여야 할것이다. 게다가 공격자가 통과암호파일의 복사를 얻을수 있으면 크랙커프로그람을 여유 있게 다른 콤퓨터우에서 실행할수 있다. 이것은 공격자가 적당한 기간에 수천개의 가능한 통과암호들을 실행할수 있게 한다.

례로서 통과암호크랙커는 1993년 8월[MADS93]에 인터네트에서 나타났다. 사고하는 콤퓨터회사(Thinking Machines Corporation)의 병렬콤퓨터를 리용하면 1초에 벡토르단위당 1560번의 암호화를 진행할수 있다. 처리마디당 4개의 벡토르를 단위로 가지는 경우 128-마디콤퓨터에서는 초당 80만번의 암호화를 진행할수 있으며 1024마디를 가지는 콤퓨터에서는 초당 640만번의 암호화를 진행할수 있다.

이렇게 추측률은 높지만 아직은 공격자가 통과암호를 발견하기 위해 기호들의 가능한 모든 조합을 해보는 우둔하고 힘내기공격방법에 의거할 정도는 아니다. 그대신 통과암호크랙커들은 일부 사람들이 쉽게 추측할수 있는 통과암호를 선택한다는 사실에 기대를 건다.

일부 사용자들은 자기의 통과암호를 선택할 때 무의미하게 짧은것을 선택한다. 퍼듀(Purdue)종합대학에서 연구한 결과를 표 15-1에 보여 주었다. 연구에서는 약 7000명의 사용자들의 등록자리가 나타나는 54대의 콤퓨터에서 통과암호를 선택할 때 일어 나는 변

화들을 관찰하였다. 통과암호의 거의 3%는 길이가 3문자이하의 문자렬이였다. 공격자는 길이가 3이거나 이보다 작은 모든 통과암호들을 모조리 검사하는 방법으로 공격해 볼수 있다. 이와 같은 공격을 방어하는 간단한 방법은 체계가 6문자이하의 통과암호를 선택하는것을 거부하거나 심지어 모든 통과암호의 길이가 정확히 8문자이도록 요구하는것이다. 대부분의 사용자들은 이런 제한에 대하여 불만을 가지지 않을것이다.

표 15-1. 관측된 통과암호길이[SPAF92n]

길이	개수	비률
1	55	.004
2	87	.006
3	212	.02
4	449	.03
5	1260	.09
6	3035	.22
7	2917	.21
8	5772	.42
계	13787	1.0

통과암호의 길이는 보안문제의 일부분일뿐이다. 많은 사람들은 통과암호를 선택할 때 자신의 이름,자신이 살고 있는 거리이름,보통 쓰는 사전의 단어 등과 같이 추측할수 있는 통과암호를 선택한다. 이것은 쉽게 통과암호를 파괴할수 있게 한다. 크랙커는 통과암호일수 있는 목록에 대해서 통과암호파일을 단순히 검사만 하면 될수 있다. 많은 사람들이 추측할수 있는 통과암호를 리용하므로 이런 방법은 사실상 거의 모든 체계에서 성공할것이다.

추측의 효과성에 대한 한가지 실례가 참고문헌 [KLEI90]에 제시되였다. 저자는 각이한 출처로부터 약 14000개의 암호화된 통과암호들이 들어 있는 UNIX의 통과암호파일들을 수집하였다. 저자가 정확히 위협으로 특징 지은 결과를 표 15-2에 보여 주었다. 통과암호의 거의 4분의 1가량이 추측되였다. 다음과 같은 추측전략을 리용하였다.

1. 사용자의 이름,아명,등록자리이름,기타 관련이 있는 개인정보를 입력시켜 본다. 매 사용자에 대해서 모두 130개의 각이한 조합을 입력시켜 보았다.
2. 각이한 사전들의 단어를 입력시켜 본다. 저자는 체계자체의 직결식사전과 표에 보여 준 기타 여러 목록들을 비롯하여 6만개이상의 단어를 콤파일하였다.
3. 우의 단계 2에서 얻은 단어들에 대한 각이한 조합을 입력시켜 본다. 이 조합에는 첫번째 문자가 대문자이거나 조종기호인것,전체 단어가 대문자인것,거꾸로 쓴 단어, 문자《o》을 수자《0》으로 변화시킨것 등이 속한다. 이 조합은 입력시켜야 할 목록에 새로 100만개의 단어를 추가한다.
4. 단계 3에서 고려되지 않은 단계 2의 단어들에 대한 각이한 대문자조합을 입력시켜 본다. 이것은 목록에 거의 200만개의 단어를 더 추가한다.

표 15-2. 13797개의 등록자리의 표본모임에서 파괴된 통과암호[KLEI90]

통과암호의 형태	탐색규모	대조된 개수	대조된 통과암호의 퍼센트	비용/리득률
사용자/등록자리이름	130	368	2. 7%	2. 830
문자렬	866	22	0. 2%	0. 025
번호	427	9	0. 1%	0. 021
중국어	392	56	0. 4%	0. 143
지역이름	628	82	0. 6%	0. 131
일반이름	2239	548	4. 0%	0. 245
녀자이름	4280	161	1. 2%	0. 038
남자이름	2866	140	1. 0%	0. 049
드문이름	4955	130	0. 9%	0. 026
신화와 전설	1246	66	0. 5%	0. 053
쉑스피어작품	473	11	0. 1%	0. 023
체육용어	238	32	0. 2%	0. 134
과학허구	691	59	0. 4%	0. 085
영화와 배우	99	12	0. 1%	0. 121
만화	92	9	0. 1%	0. 098
명인	290	55	0. 4%	0. 190
성구와 형식	933	253	1. 8%	0. 271
성(이름)	33	9	0. 1%	0. 273
생물학	58	1	0. 0%	0. 017
체계사전	19683	1027	7. 4%	0. 052
콤퓨터이름	9018	132	1. 0%	0. 015
기억술	14	2	0. 0%	0. 143
King James Bible	7525	83	0. 6%	0. 011
잡다한 단어	3212	54	0. 4%	0. 017
이디슈어(Yiddish)	56	0	0. 0%	0. 000
소행성	2407	19	0. 1%	0. 007
계	62727	3340	24. 2%	0. 053

이렇게 약 300만개의 단어를 검사해야 한다. 앞에서 설명한 고속의 사고하는 콤퓨터를 리용하면 가능한 모든 쏠트값에 대해서 이 모든 단어들을 암호화하는데 걸리는 시간은 1시간이하이다. 이와 같은 완전탐색은 약 25%의 비률로 성공할수 있다는것을 알아 두시오. 그러나 단번에 체계의 넓은 령역에 대한 특권을 충분히 얻을수도 있다.

접근조종

통과암호에 대한 공격을 막는 한가지 방법은 통과암호파일에 대한 적수의 접근을 거부하는것이다. 암호화된 통과암호파일의 구역에 특권사용자만이 접근할수 있다면 적수는 특권사용자의 통과암호를 미리 알지 못하고서는 그것을 읽을수 없다. 참고문헌 [SPAF92a]에서는 이 전략에 대한 몇가지 결함을 지적하였다.

- 거의 모든 UNIX체계들을 비롯한 많은 체계들은 불의의 침입에 대해서 민감하다. 공격자는 일단 어떤 방법으로 접근자격을 얻으면 검출될 위험이 적어 지도록 다

른 가입대화용의 등록자리들을 리용하기 위하여 통과암호들을 수집하려 할수 있다. 또한 등록자리를 가진 사용자는 특권자료에 접근하거나 체계를 파괴하기 위해 다른 사용자의 등록자리를 가지려고 할수 있다.
- 보호사고는 통과암호파일을 읽을수 있게 함으로써 모든 등록자리들을 위태롭게 한다.
- 어떤 사용자들은 다른 콤퓨터상의 보호령역에 등록자리를 가지고 같은 통과암호를 리용할수 있다. 그러므로 어느 한대의 콤퓨터에서 그 누군가가 통과암호를 읽을수 있으면 다른 곳에 있는 콤퓨터는 위태롭게 될수 있다.

따라서 보다 효과적인 전략은 사용자가 추측하기 힘든 통과암호를 선택하는것이다.

통과암호선택전략

표 15-1과 표 15-2에 서술한 두 실험의 결과는 많은 사용자들이 너무 짧거나 쉽게 추측할수 있는 통과암호를 선택한다는것이다. 기껏해서 통과암호로서 사용자들이 눈으로 볼수 있는 8개의 기호들을 란수적으로 선택한다면 통과암호를 효과적으로 크랙킹할수 없다. 그러나 이때 대부분의 사용자들이 자기들의 통과암호를 기억하는것은 거의 불가능할것이다. 기억할수 있는 적당한 기호렬들로 통과암호의 령역을 제한한다고 해도 다행히도 이 령역의 규모는 실제적으로 크랙킹하기에는 매우 크다. 이때 목표는 사용자가 기억할수 있는 통과암호를 선택할 때 추측할수 있는 통과암호를 제거하는것이다.

다음의 4가지 기본방법이 리용되고 있다.

- 사용자교육
- 콤퓨터로 생성된 통과암호
- 통과암호의 역검사
- 밀기식통과암호의 검사

사용자교육전략은 사용자들에게 추측이 힘든 통과암호를 리용하는것이 가지는 중요성을 강조하며 강한 통과암호를 선택하기 위한 지침을 주는것이다. 이러한 전략은 특별히 사용자수가 많으며 사용자들이 자주 변화될 때는 적합하지 않을수 있다. 많은 사용자들은 이 지침을 단순히 무시할것이다. 또 어떤 사람들은 어느것이 강한 통과암호이라는것을 잘 판단 못할수 있다. 례를 들어 많은 사용자들은 단어를 반대로 쓰거나 마지막문자를 대문자로 쓰면 통과암호를 추측할수 없을것이라고 잘못 생각하고 있다.

콤퓨터로 생성된 통과암호들도 문제가 있다. 통과암호가 진짜 완전한 우연수이라면 사용자들은 그것을 기억할수 없을것이다. 통과암호가 만들어 졌다 해도 사용자가 그것을 기억하기 어렵고 적을 때 혼돈될수 있다. 일반적으로 사용자는 콤퓨터로 생성된 통과암호도식을 잘 리용하지 않는다. FIPS PUB 181에서는 잘 설계된 통과암호자동생성기를 정의하고 있다. 표준생성기는 방식의 설명서뿐아니라 알고리듬을 C언어로 작성한 원천부호의 완전한 목록도 가지고 있다. 알고리듬은 읽을수 있는 말마디들을 만들고 그것들을 단어가 되도록 련결함으로써 단어를 생성한다. 란수생성기는 말마디들과 단어들을 조립하여 우연적인 기호렬을 만든다.

막기식통과암호검사전략은 체계가 추측될수 있는 통과암호를 찾아 내기 위해 자기의 통과암호크랙커를 정기적으로 실행시키는 전략이다. 체계가 추측되여 사용자에게 통지된 통과암호들은 모두 제거된다. 이 전술에는 몇가지 결함들이 있다. 첫째로, 일감을

정확히 수행하려면 자원을 집중적으로 소비한다는것이다. 통과암호파일을 훔칠수 있는 단호한 적수가 몇시간 지어 며칠동안 통과암호를 추측하기 위해 CPU시간을 완전히 소비할수 있기때문에 효과적인 막기식통과암호검사자도 명백히 불리하다. 더우기 체계에 존재하는 임의의 통과암호들은 막기식통과암호검사자가 그것을 검사하기전에는 약한 통과암호로 남아 있다.

밀기식통과암호검사전략은 통과암호의 보안을 개선하는 가장 전망적인 방식이다. 이 방식에서는 사용자자신이 통과암호를 선택하게 한다. 그러나 선택할 때에 체계는 통과암호가 허용되는가를 검사하고 허용되지 않으면 그것을 거부한다. 이와 같은 검사는 체계로부터 충분한 지도를 받으며 사전식공격으로 추측할수 없는 매우 큰 통과암호공간으로부터 사용자들이 기억할수 있는 통과암호를 선택할수 있다는 원리에 기초하고 있다.

밀기식통과암호검사전략은 사용자의 허용성과 강도사이의 균형을 맞추는것이다. 체계가 너무 많은 통과암호들을 거부하면 사용자들은 통과암호를 선택하기가 너무 힘들다고 불평할것이다. 체계가 허용할수 있는 통과암호를 정의하는 어떤 단순한 알고리듬을 리용하면 이것은 통과암호크랙커들에게 추측기술을 세련시킬수 있는 방법을 제공한다. 아래에서 밀기식통과암호를 검사하는 방식들을 고찰하자.

첫번째 방식은 규칙을 실시하는 단순한 체계를 리용하는 방식이다. 례를 들어 다음의 규칙을 실시할수 있다.

- 모든 통과암호들은 적어도 8문자길이여야 한다.
- 통과암호들의 첫 8개의 문자들중에는 적어도 하나의 대문자나 소문자,수자와 구두점기호가 포함되여야 한다.

이러한 규칙들에는 사용자에 대한 충고가 결합될수 있다. 이 방식은 단순히 사용자를 교육하는데서는 우월하지만 통과암호크랙커를 막는데는 불충분할수 있다. 이 방식은 크랙커들에게 어떤 통과암호에 대해서는 크랙킹하지 말것을 경고하지만 통과암호를 크랙킹할수 있는 가능성은 여전히 있다.

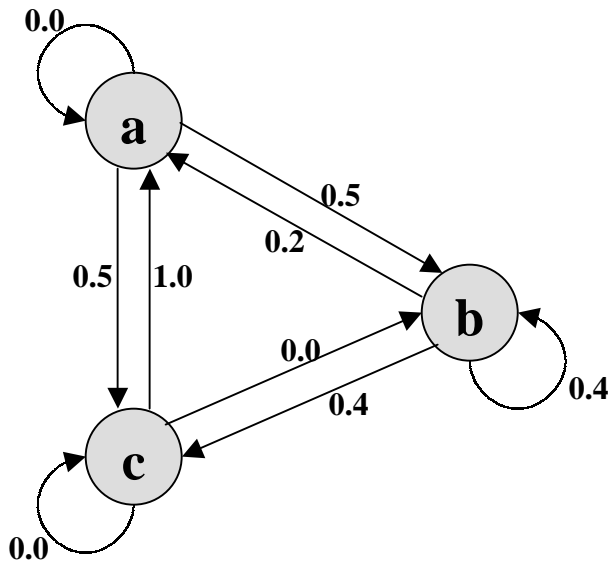
또 다른 한가지 방식은 가능한 《나쁜》 통과암호들로 이루어 진 큰 사전을 편집하는것이다. 사용자가 통과암호를 선택할 때 체계는 그것이 허용되지 않는 목록에 없는가를 확인하기 위해 통과암호를 검사한다. 이 방식에는 두가지 문제가 있다.

- **공간:**이 방식이 효과적이기 위해서는 사전이 매우 커야 한다. 례를 들어 퍼듀(Purdue)의 연구에서 리용한 사전은 30MB이상의 기억공간을 차지한다[SPAF92a]).
- **시간:**큰 사전을 탐색하는데 요구되는 시간은 길다. 게다가 사전단어들의 적당한 치환에 대해 검사하기 위해서는 이 단어들의 대부분이 사전에 포함되므로 실로 큰 사전을 만들든가 혹은 매 탐색은 많은 처리를 필요로 하게 된다.

약속된 목록의 단어들을 제거하는 방식에 기초한 효과적이고 능률적인 밀기식통과암호검사자를 개발하는데는 두가지 기술이 있다. 하나는 추측할수 있는 통과암호들을 생성하는 마르꼬브모형을 개발하는것이다[DAVI93]. 그림 15-2에 이와 같은 모형을 간단히 제시하였다. 이 모형은 3문자자모로 이루어 진 언어로 표시된다. 어떤 순간에 체계의 상태는 가장 최근의 문자와 일치한다. 한 상태로부터 다른 상태에로의 이행값은 어떤 문자가 다른 문자로 될 확률이다. 따라서 현재 문자가 a일 때 다음 문자가 b일 확률은 0.5이다.

일반적으로 마르꼬브모형은 4인수조$[m, A, T, k]$이며 여기서 m은 모형에서의 상태수이고 A는 상태공간, T는 이행확률행렬, k는 모형의 차수이다. k차모형에 관해서 개개의 문자에로 이행할 확률은 이전에 생성된 k개의 문자에 관계된다. 그림 15-2에 간단한 1차모형을 보여 주었다.

다음은 2차모형의 개발과 리용과 관련한 문제이다. 먼저 추측할수 있는 통과암호사전을 구성한다. 다음 아래와 같은 방법으로 이행행렬을 구한다.

1. 빈도수행렬 f를 결정한다. 여기서 f$(i,\ j,\ k)$는 각각 i, j, k번째 문자로 이루어진 세 글자조가 나타날 회수이다. 례를 들어 통과암호 *parsnips*는 세 글자모임인 par, ars, rsn, sni, nip, ips를 생성한다.
2. 매 두 글자조인 ij에 대해서는 ij로 시작하는 세 글자조의 전체 개수로서 f$(i,\ j,\ \infty)$를 계산한다. 례를 들어 f(a, b, ∞)는 aba, abb, abc 등 형식의 세 글자조의 전체 개수일것이다.
3. T요소의 계산은 다음과 같다. $T(i,j,k) = \dfrac{f(i, j, k)}{f(i, j, \infty)}$

M={3, {a, b, c}, T, 1}

여기서 $T=\begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.4 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$

이 언어로부터 확률적으로 생성될수 있는 기호렬:abbcacaba

이 언어로부터 확률적으로 생성될수 없는 기호렬:aacccbaaa

그림 15-2. 마르꼬브모형의 실례

마르꼬브모형의 결과는 사전에 있는 단어들의 구조를 반영하는 모형이다. 이 모형에서 질문《이것은 나쁜 단어인가?》는 질문《이 문자렬(통과암호)은 이 마르꼬브모형에 의해 생성되였는가?》로 변환된다. 주어 진 통과암호에 대한 모든 세 글자조의 이행확률을 찾을수 있다. 통과암호가 이 모형에 적합한가 적합하지 않은가를 결정하는데 일부 표준통계적검사들을 리용할수 있다. 모형에 의해서 생성될수 있는 통과암호들은 거절된다. 2차모형을 리용하면 좋은 결과들을 얻을수 있다. 이 모형들을 리용한 체계는 사전에서 사실상 모든 통과암호들을 발견하며 따라서 사용자친숙형이 아닌 적지 않은 좋은 통과암호들도 배제하지 않는다.

스패호드(Spafford)(참고문헌 [SPAF92a, SPAF92b])가 제안한 다른 방식도 있다. 이것은 블룸(Bloom)려과기(참고문헌 [BLOO70])를 리용하는데 기초하고 있다. 먼저 블룸려과기의 조작을 설명하자. k차블룸려과기는 k개의 독립적인 하쉬함수 $H_1(x), H_2(x), \cdots, H_k(x)$의 모임으로 이루어 진다. 여기서 매 함수는 통과암호를 0부터 (N-1)구간의 하쉬값으로 넘기는 함수이다. 즉

$$H_i(X_j)=y \qquad 1 \le i \le k; \qquad 1 \le j \le D; \qquad 1 \le y \le N-1$$

여기서

X_j :통과암호사전에서 j번째 단어

D :통과암호사전에서 단어의 수

다음의 절차를 사전에 적용한다.

1. N개 비트들의 하쉬표의 모든 비트들을 초기에 0으로 설정한다.
2. 매 통과암호에 대해 k개의 하쉬값을 계산하여 하쉬표에서 대응하는 비트를 1로 설정한다. 실례로 어떤 (i, j)에 대하여 $H_i(x_j)=67$이면 하쉬표의 67번째 비트는 1로 설정된다. 비트가 이미 값 1로 설정되여 있으면 그대로 한다.

새로운 통과암호가 검사자에게 제출되면 그것의 k개의 하쉬값을 계산한다. 만일 하쉬표에서 대응하는 모든 비트들이 1이면 통과암호는 거부된다. 이때 사전에 있는 모든 통과암호들이 거부될수도 있다. 그러나 여기에는 일부 《잘못된 긍정》으로 된것들이 있을수 있다(이것은 사전에는 없지만 하쉬표에서 배후자를 선택한다). 이것을 보기 위해 두개의 하쉬함수를 가지는 도식을 생각한다. 통과암호 *undertaker*와 *hulkhogan*은 사전에 있지만 *xG%#jj*98은 없다고 가정한다. 또한

$$H_1(\text{undertaker}) = 25 \qquad H_1(\text{hulkhogan}) = 83 \qquad H_1(\text{xG\%\#jj98}) = 665$$
$$H_2(\text{undertaker}) = 998 \qquad H_2(\text{hulkhogan}) = 665 \qquad H_2(\text{xG\%\#jj98}) = 998$$

이라고 가정한다.

통과암호 xG%#jj98이 체계에 제출되면 비록 그것이 사전에 없다고 할지라도 거부될것이다. 이러한 잘못된 긍정들이 너무 많으면 사용자가 통과암호를 선택하기가 힘들것이다. 따라서 잘못된 긍정이 최소로 되는 하쉬도식을 설계하는것이 좋다.

잘못된 긍정의 확률을 식

$$P \approx (1-e^{-kD/N})^{k} \approx (1-e^{-k/R})^{k}$$

으로 대략 계산할수 있다. 혹은 동등하게

$$R \approx \frac{-k}{\ln(1-P^{1/k})}$$

여기서

k : 하쉬함수의 개수
N : 하쉬표에서 비트들의 수
D : 사전에서 단어들의 수
R : N/D–사전크기(단어수)에 대한 하쉬표크기(비트수)의 비

그림 15-3은 각이한 k값에 대한 P와 R사이의 관계를 보여 준다. 100만개의 단어가 있는 사전이 있고 사전에 없는 통과암호를 거부할 확률이 0.01이라고 가정한다. 6개의 하쉬함수를 선택하면 요구되는 비는 R=9.15이다. 따라서 9.6×10^{6}bit 혹은 약 1.2MB의 하쉬표가 필요하다. 이에 비해 총체적인 사전의 기억공간으로 약 8MB가 필요하다. 따라서 거의 7배로 압축할수 있다. 또한 통과암호검사는 6개의 하쉬함수에 대한 간단한 계산으로 되며 사전의 크기에 관계 없다. 사실상 사전을 다 리용하는것이 진짜 탐색이다.

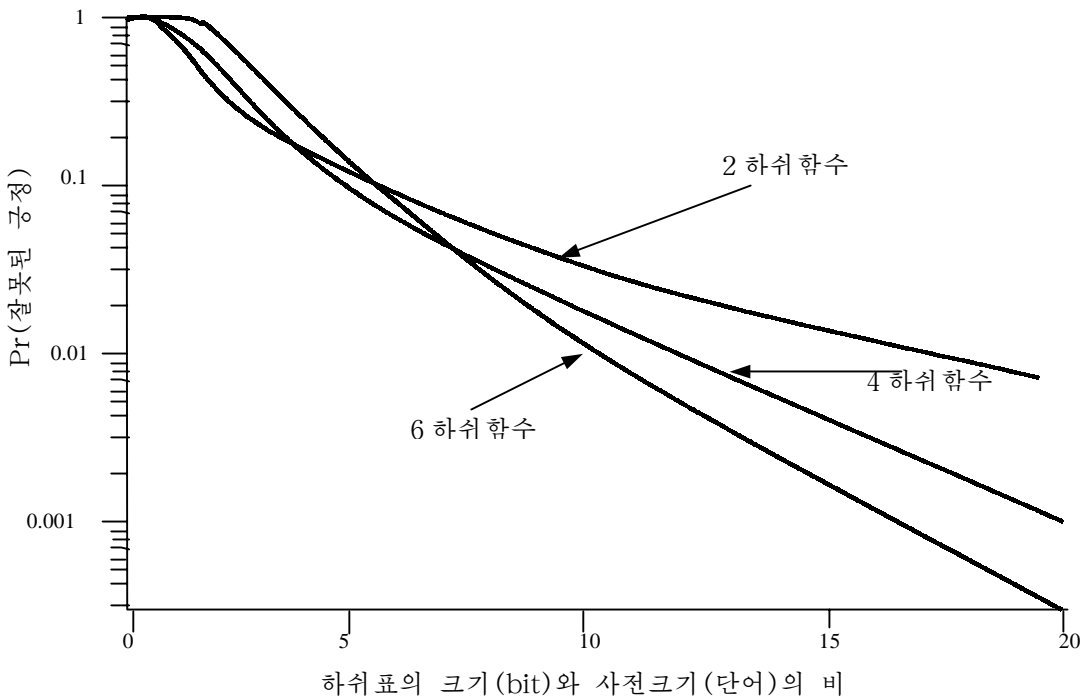



그림 15-3. 블룸려파기의 실행

침입검출

아무리 훌륭한 침입방어체계라도 실수할수 있다. 체계의 두번째 방어선은 침입검출로서 최근에 많은 연구들에서 초점으로 되고 있다.

그 리유를 보면 다음과 같다.

1. 침입을 즉시 검출하였다면 어떤 피해를 받거나 자료가 손상되기전에 침입자를 확인하여 체계에서 제거할수 있다. 침입자를 앞질러 충분히 제때에 검출하지 못한다 해도 침입을 더 빨리 검출하면 할수록 손해량은 더 적으며 더 빨리 회복할 수 있다.
2. 효과적인 침입검출체계는 침입을 막는 방어와 같이 봉사할수도 있다.
3. 침입검출은 침입방어를 강화하는데 리용할수 있는 침입기술에 대한 정보를 수집할수 있게 한다.

침입검출은 정량적방법에서 침입자의 행동이 합법적인 사용자의 행동과는 차이난다는 가정에 기초하고 있다. 물론 침입자의 공격과 합법적인 사용자의 표준적인 자원리용사이에 뚜렷하고 정확한 차이가 있을수 있다고 기대하기는곤난하다. 오히려 공통성이 있을수 있다고 보아야 한다.

그림15-4는 침입검출체계의 설계자에게 과제의 본질을 매우 추상적으로 암시해 준다. 침입자의 일반적인 행동이 합법적인 사용자의 일반적인 행동과 다르다 하더라도 이 행동들에는 일치되는것이 있다. 따라서 더 많은 침입자들을 발견할수 있는 침입자의 행동에 대한 애매한 해석은 《잘못된 긍정》이 많아 지게 즉 합법적인 사용자들을 침입자로 보게 할수도 있다. 다른 한편 침입자의 행동에 대한 잘 째인 해석으로 잘못된 긍정을 제한하는것은 《잘못된 부정》을 증가하게 한다. 즉 침입자를 침입자로 보지 않게 할수 있다. 따라서 침입을 검출하는 실천에서는 타협과 숙련의 두 요소가 필요하다.

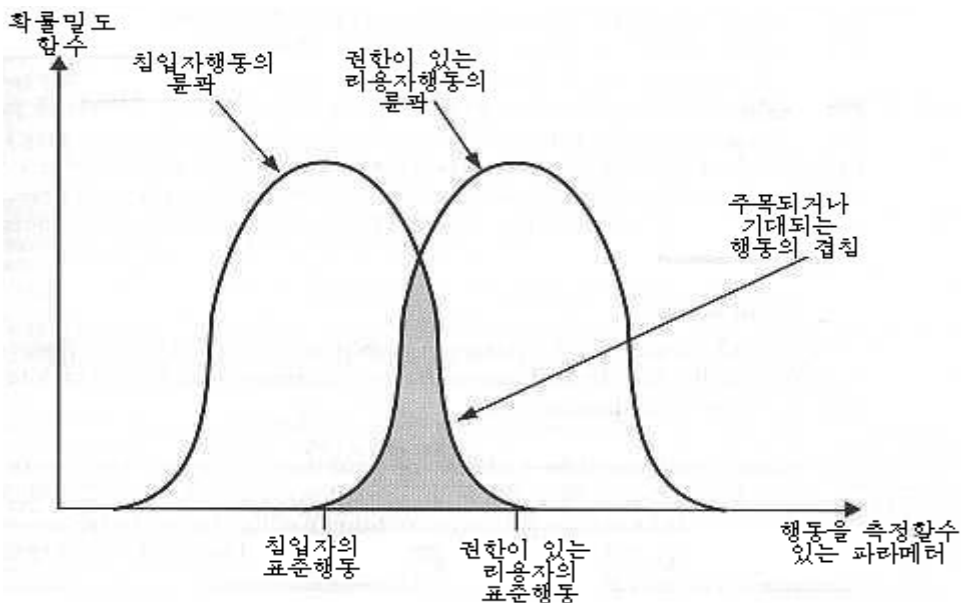



그림 15-4. 침입자와 권한이 있는 사용자의 행동의 륜곽

앤더슨(Anderson)의 연구(참고문헌[ANDE80])에서는 누구든 정당한 근거를 가지고 가장자와 합법적인 사용자를 구별할수 있다고 가정하였다. 합법적인 사용자의 행동에 대한 패턴은 과거의 리력을 관찰함으로써 확립할수 있으며 이런 패턴들사이의 유효편차는 검출할수 있다. 앤더슨은 정상적인 행동과 비정상적인 행동의 차이는 작을수 있으므로 불법행위자(권한을 부여 되지 않은 방식에서 행동하는 합법적인 사용자)를 검출하는 과제가 더 힘들다는것을 제의하였다. 앤더슨은 이런 침입은 단순히 례외적인 행동에 대한 탐색을 통해서는 검출할수 없다고 결론하였다. 그럼에도 불구하고 권한이 없이 리용하는 행동을 나타내는 조건클라스에 대한 지적인 정의에 의해 불법행위자의 행동을 검출할수 있다. 결국 도용자의 검출은 순수한 자동화기술의 범위를 벗어 나게 되였다. 1980년에 진행된 이러한 관찰들은 오늘날 사실로 되였다.

참고문헌[PORR92]에서는 침입검출에 대한 다음의 방식을 확인하였다.

1. **통계적인 비정상검출**: 합법적인 사용자의 한주기동안의 행동과 관련한 자료를 수집한다. 다음 합법적인 사용자의 행동이 아닌가를 높은 정확도로 결정하기 위해 관찰되는 행동에 통계적검사들을 적용한다.
   ㄱ) 한계검출:이 방식은 각이한 사건발생의 빈도수에 대한 한계 즉 사용자의 독립성의 정의를 포함한다.
   ㄴ) 륜곽에 기초한것:개별적인 등록자리들의 행동변화를 검출하기 위해 매 사용자의 활동륜곽을 개발하고 리용한다.
2. **규칙에 기초한 검출**:주어 진 행동이 침입자의 행동이라는것을 결정하는데 리용할수 있는 규칙들의 모임을 정의하는 방식이다.
   ㄱ) 비정상검출:이전에 사용한 패턴으로부터의 편차를 검출하기 위한 규칙들을 개발한다.
   ㄴ) 침입식별:의심스러운 행동을 탐색하는 전문가체계방식이다.

한마디로 말하여 통계적방식은 정상이거나 기대하는 행동을 정의하려고 하는것이고 규칙에 기초한 방식은 고유한 행동을 정의하려고 하는 방식이다.

앞부분에서 서술한 공격자들의 류형들중 통계적인 비정상검출은 등록자리의 고유한 행동패턴을 위조하지 않는 가장자에 대하여 효과적이다. 다른 한편 이런 기술로는 불법행위자에 대해서 론의할수 없다. 이런 공격에 대하여서는 규칙에 기초한 방식이 침입을 밝히는 사건과 렬들을 식별할수 있다. 실지로 체계에서는 일반적인 공격에 대하여 효과성을 높이기 위해 두 방식을 결합할수 있다.

검열기록

침입을 검출하는 기본도구는 검열기록이다. 활동중에 있는 어떤 기록은 침입검출체계의 입력자료로서 사용자에 의해 보존되여야 한다. 기본적으로 두가지 계획을 리용한다.

- **가공하지 않은 검열기록:**실제상 모든 다중사용자조작체계는 사용자의 활동에 대한 정보를 수집하는 등록자리쏘프트웨어를 포함한다. 이 정보를 리용하는 우점은 보충적인 수집쏘프트웨어가 필요 없다는것이다. 결함은 가공하지 않은 검열기록이 필요한 정보를 가지고 있지 않거나 혹은 편리한 형태로 가지고 있지 못할수도 있다는것이다.
- **검출용검열기록:**수집기구는 침입검출체계가 요구하는 정보만을 포함하는 검열기록들을 생성할수 있다. 이런 방식의 한가지 우점은 판매자에게 무관계하며 각이

한 종류의 체계에 입력할수 있다는것이다. 결함은 보건대 체계에서 실행되는 두 개의 등록자리쏘프트웨어를 가지는데 드는 간접비용이 있다는것이다.

검출용검열기록의 좋은 실례로는 도로티 덴닝(DorotyDenning)(참고문헌 [DENN87])이 개발한것을 들수 있다. 매 검열기록는 다음의 마당을 포함한다.

- **주동체:** 작용의 초기자이다. 주동체는 일반적으로 말단사용자나 사용자그룹을 대신하여 작용하는 처리공정이 될수도 있다. 모든 활동은 주동체가 일으키는 지령을 통해 발생한다. 주동체들은 서로 다른 접근클라스들로 그룹화될수 있으며 이 클라스들은 겹칠수 있다.
- **작용:** 조작은 주동체나 대상에 의해 수행된다. 례를 들어 가입,읽기,I/O수행,실행이다.
- **객체:** 작용의 접수자이다. 실례를 들어 파일, 프로그람, 통보문, 기록, 말단, 인쇄기, 사용자 혹은 프로그람창조도구 등이다. 주동체는 전자우편과 같이 작용을 접수한 다음 대상을 고려한다. 대상은 형에 따라 그룹화될수 있다. 대상은 대상형이나 환경에 따라 변화될수 있다. 례를 들어 자료기지의 작용들은 전체 혹은 기록준위에서 자료기지를 검열할수 있다.
- **례외조건:**. 어떤 례외조건이 발생하면 탈퇴하겠는가이다.
- **자원사용법:**매 요소가 리용한 자원량(즉 인쇄하거나 현시한 행의 수,읽거나 쓴 기록의 수,처리시간,리용한 I/O단위,대화시간 등)을 주는 량적인 요소목록이다.
- **시간도장:**언제 작용이 수행되였는가를 확인하는 유일한 날자-시간도장이다.

대부분의 사용자조작들은 여러개의 작용들로 이루어 진다. 례를 들어 파일복사는 사용자의 지령의 실행이며 이것은 한개의 파일로부터 읽기,다른 파일에로의 쓰기를 비롯하여 접근의 확인과 복사의 설정을 포함한다. 스미스(Smith)가 만든 지령을 고찰하자.

COPY GAME. EXE TO <Library>GAME. EXE

이 지령은 현재의 등록부로부터 <Library>등록부에로 실행형파일GAME을 복사한다. 다음과 같은 검열기록들이 생성될수 있다.

Smith	실행	<Library>COPY. EXE	0	CPU = 00002	11058721678
Smith	읽기	<Smith>GAME. EXE	0	RECORDS = 0	11058721679
Smith	실행	<Library>COPY. EXE	Write-viol	RECORDS = 0	11058721680

이 경우에 복사는 스미스가 <Library>등록부에로의 쓰기허락을 받지 못하였으므로 실패한다.

사용자조작을 기초적인 작용으로 분해하는것은 세가지 우점을 가진다.

1. 대상들은 체계에서 보호가능한 실체들이므로 기초적인 작용을 리용하여 대상에 작용하는 모든 행동을 검열할수 있다. 그러므로 체계는 접근조종을 파괴하려는 것을 검출할수 있으며(되돌려 지는 례외조건들의 수에서 비정상적인것에 주목함

으로써) 주동체에 접근할수 있는 대상모임에서 비정상적인것에 주목하면서 성공한 파괴를 검출할수 있다.

2. 단일대상,단일작용검열기록들은 모형작성과 실현이 간단하다.
3. 검출용검열기록들은 간단하고 단일한 구조이므로 이미 존재하는 가공하지 않은 검열기록으로부터 검출용검열기록에로 직접 넘김으로써 검열정보나 최소한 그 일부분을 구하는것이 상대적으로 쉬울수 있다.

통계적비정상검출

우에서 지적한바와 같이 통계적비정상검출기술은 크게 두가지 부류 즉 한계검출과 륜곽에 기초한 체계로 이루어 져 있다. 한계검출은 어떤 시구간우에서 특정한 사건형태의 발생수를 계산한다. 계산한 값이 발생할수 있을것이라고 생각하는 적당한 수를 넘어나면 침입이라고 가정한다.

한계검출만으로는 완전하지 못하며 일정하게 복잡한 공격을 검출하는것도 효과적이지 못하다. 한계와 시구간을 둘다 결정하여야 한다. 사용자들의 가변성때문에 이런 한계는 잘못된 긍정이든가 아니면 잘못된 부수를 많이 생성할수 있다. 그러나 단순한 한계검출은 더 정교한 기술과 결합되면 쓸모 있다.

륜곽에 기초한 비정상검출은 개별적인 사용자들이나 서로 관계되는 사용자들의 그룹의 과거행동을 특징 지은 다음 주요편차의 검출에 주목한다. 륜곽은 파라메터의 모임으로 구성할수 있으므로 단일파라메터의 편차는 근본적으로 경보신호를 발생하는데 충분하지 못할수 있다.

이 방식의 기초는 검열기록의 해석이다. 검열기록는 두가지 방법으로 침입검출함수에 입력을 제공한다. 첫째로 설계자는 사용자의 행동을 측정하는데 리용할수 있는 량적인 측정수단을 결정해야 한다. 시간주기우에서 검열기록을 해석함으로써 일반사용자의 활동륜곽을 결정할수 있다. 그러므로 검열기록는 일반적인 행동을 정의한다. 둘째로 침입을 검출하기 위해 현재의 검열기록을 입력자료로 리용한다. 이것은 침입검출모형이 평균행동과의 편차를 결정하기 위해 수입검열기록을 해석한다는것이다.

륜곽에 기초한 침입검출에 쓸모 있는 측정수단의 실례는 다음과 같다.

- **계수기:**관리작용에 의해 재설정될 때까지 증가는 할수 있지만 감소하지 않는 비부옹근수이다. 일반적으로 어떤 사건형태의 계수는 특정한 시간주기우에서 진행한다. 실례로서 한시간동안에 단일사용자가 가입한 개수,단일사용자대화동안 실행하는 주어 진 지령의 회수,1분동안 실패하는 통과암호의 개수를 들수 있다.
- **표준규격:**증가하거나 감소할수 있는 비부옹근수이다. 일반적으로 표준규격은 어떤 실체의 현재상태의 값을 측정하는데 리용된다. 실례로 사용자의 응용프로그람에 할당된 국부적인 접속수,사용자의 처리공정이 발생한 통보문의 수를 들수 있다.
- **구간시계:**두개의 관계되는 사건들사이의 시간길이이다. 실례로 성공적으로 가입한 등록자리들사이의 길이이다.
- **자원리용상태:**특정한 주기동안 소비하는 자원의 량이다. 실례로 사용자의 대화동안 인쇄한 페지수, 프로그람을 실행하는데 소비한 전체 시간을 들수 있다.

이러한 일반적인 측정기술이 주어 졌을 때 현재활동이 접수가능한 한계내에 있는가를 결정하기 위해 여러가지 검사들을 진행할수 있다. 참고문헌[DENN87]에서는 다음의

검사방식을 서술한다.

- 평균값과 표준편차
- 다변량
- 마르꼬브공정
- 시계렬
- 조작

가장 단순한 정적검사는 어떤 리력주기에서 파라메터의 **평균값과 표준편차**를 측정하는것이다. 이것은 일반적인 행동과 이것의 가변성을 나타낸다. 평균값과 표준편차는 계수기,시계,자원측정과 같은 여러가지 측정수단으로 측정할수 있다. 그러나 이러한 측정들자체는 침입검출목적에 비해 볼 때 일반적으로 아직 완성되지 못하였다.

다변량모형은 둘 혹은 그이상의 변수들사이의 련관성에 기초하고 있다. 침입자의 행동은 이와 같은 련관성을 고려하여 보다 큰 확실성으로 특징 지을수 있다(실례로 처리기시간과 자원사용이나 가입빈도수,대화시간들사이의 련관성을 들수 있다).

마르꼬브처리공정모형은 여러 상태들의 이행확률을 표현하는데 사용한다. 례를 들어 이 모형은 어떤 지령들사이의 이행을 고찰하는데 리용할수 있다.

시계렬모형은 너무 빠르게 혹은 너무 느리게 일어 나는 사건들의 렬을 관찰하면서 시구간에 주목한다. 각이한 통계적검사들은 비정상적인 동기를 특징 짓는데 리용할수 있다.

마지막으로 **조작모형**은 과거의 검열기록의 자동적인 해석보다도 비정상적인것으로 고찰되는것들에 대한 판단에 기초하고 있다. 일반적으로 고정된 한계를 정의하며 침입은 이 한계밖에 있다고 추측한다. 이러한 방식은 어떤 형태의 활동으로부터 침입자의 행동을 추출하는데 가장 효과적으로 적용된다. 례를 들어 짧은 시간동안 가입을 많이 해보는것은 침입자이다.

이러한 여러가지 측정과 모형을 리용하는 실례로서 표15-3에 스탬포드연구소(SRI)의 침입검출체계(IDES)를 참고하고 검사한 여러가지 측정을 보여 주었다(참고문헌[DENN87, JAVI91, LUNT88]).

통계적륜곽을 리용하는 기본 우점은 보안결함에 대한 사전지식을 요구하지 않는다는것이다. 검출프로그람은 무엇이 《정상적인》행동인가를 알고 다음 편차를 구한다. 이 방식은 체계의 특징들과 약점들에 관계되지 않는다. 그러므로 각이한 체계에 실지로 적용할수 있을것이다.

규칙기초침입검출

규칙에 기초한 기술은 체계의 사건을 관찰한 다음 규칙모임을 적용하여 주어 진 활동패턴이 의심스러운가 아닌가를 판단함으로써 침입을 검출한다. 매우 일반적인 관점에서 보면 비정상검출이나 침입식별에 주목함으로써 이 방식들에 중복이 있다고 해도 모든 방식들을 특징 지을수 있다.

규칙에 기초한 비정상검출은 방식과 강도의 측면에서 통계적비정상검출과 류사하다. 사용패턴들을 식별하고 이 패턴들을 서술하는 규칙들을 자동적으로 생성하기 위해 규칙에 기초한 방식에 따라 리력검열기록들을 분석한다. 규칙들은 사용자, 프로그람, 특권, 시간간격, 말단 등의 과거의 행동패턴을 나타낼수 있다. 현재의 행동은 그때 관찰되며 이것이 리력적으로 관찰된 어느 행동패턴과 일치하는가를 결정하기 위해 매 이행을 규칙모임에 관하여 대조한다.

표 15-3. 침입검출에 리용할수 있는 측정기술

측정	모형	검출된 침입형태
가입과 대화활동		
날자와 시간에 따르는 가입빈도수	평균값과 표준편차	침입자는 한가한 시간에 가입할수 있다.
다른 위치에서의 가입빈도수	평균값과 표준편차	침입자는 특정의 사용자가 드물게 리용하거나 전혀 리용하지 않는 위치에서 가입할수 있다.
마지막으로 가입한 때로부터의 시간	조작모형	《죽은》등록자리에로의 침입
대화당 결과시간	평균값과 표준편차	중요한 편차들은 가장자를 가리킬수 있다.
위치에로의 출력량	평균값과 표준편차	먼 곳에 많은 자료량을 전송할 때 기밀자료가 루설될수 있다.
대화자원의 리용	평균값과 표준편차	리용할수 없는 처리기나 I/O준위는 침입자를 경고할수 있다.
가입시 통과암호실패	조작모형	통과암호추측에 의한 침입시도
특정한 말단으로부터 가입의 실패	조작모형	침입을 시도한다.
지령 혹은 프로그람실행활동		
실행빈도수	평균값과 표준편차	다른 지령을 리용하거나 특권지령에 대한 접근을 가진 합법적인 사용자에 의해 성공적으로 침입하는 침입자들을 검출할수 있다.
프로그람자원의 리용	평균값과 표준편차	이상한 값은 비루스나 트로이목마의 침입을 암시할수 있으며 이것은 I/O나 처리기의 리용회수를 증가시키는 부작용을 가져 온다.
실행거부	조작모형	높은 특권준위를 가지려고 하는 개별적인 사용자들의 침입을 검출할수 있다.
파일접근활동		
읽기,쓰기,창조,삭제의 빈도수	평균값과 표준편차	개별적인 사용자들의 읽기와 쓰기접근에 대한 비정상적인것들은 가장하거나 열람하는것으로 나타날수 있다.
기록의 읽기,쓰기	평균값과 표준편차	비정상적인것은 가로채기와 집합을 통해 민감한 자료를 얻으려는 시도로 나타날수 있다.
읽기,쓰기,창조,삭제에 대한 실패회수	조작모형	권한이 없는 파일에 집요하게 접근하려고 하는 사용자들을 검출할수 있다.
파일자원의 소비계수기	조작모형	

통계적비정상검출과 같이 규칙에 기초한 비정상검출은 체계내에 어떤 보안약점을 가지고 있는가에는 관계되지 않는다. 오히려 도식은 과거의 행동의 관찰과 사실상 앞으로도 과거와 같을것이라는 가정에 기초한다. 이 방식이 효과적이기 위해서는 매우 큰 규칙자료기지가 필요하다. 례를 들어 참고문헌 [VACC89]에 서술한 도식은 10^4부터 10^6개 사이의 규칙을 포함한다.

규칙에 기초한 침입식별은 침입검출과 전혀 다른 방식을 취한다. 즉 이것은 전문가체계기술에 기초하고 있다. 이와 같은 체계의 기본특징은 알려 져 있는 침입력을 식별하거나 알려 져 있는 약점을 리용하는 침입력을 식별하기 위한 규칙들을 리용한다는것이다. 확립된 사용패턴의 한계내에 행동이 속한다 해도 규칙들을 의심이 많은 행동을 식별하도록 정의할수도 있다. 일반적으로 이러한 체계들에서 리용되는 규칙들은 콤퓨터와 조작체계에 따라 구별된다. 또한 이와 같은 규칙들은 검열기록을 자동해석하는 방법보다도 오히려 《전문가》에 의해 생성된다. 표준수속은 목표체계의 보안을 위협하는 알려 져 있는 침입씨나리오들과 기본사건들을 모두 수집하기 위해 체계관리자들과 보안분석자들이 대면하는것이다. 따라서 이 방식의 강도는 규칙들을 설정하는 기교에 관계된다.

리용할수 있는 규칙형태의 단순한 실례는 니득쓰(NIDX)에서 찾아 볼수 있는데 그것은 활동에 대한 혐의차수를 결정하는 발견적규칙들을 리용한 최초의 체계이다(참고문헌 [BAUE88]). 실례로 되는 발견적인 지식은 다음과 같다.

1. 사용자들은 다른 사용자의 개별적인 등록부들에서 파일들을 읽을수 없다.
2. 사용자들은 다른 사용자의 파일들을 쓰지 말아야 한다.
3. 근무시간후에 가입한 사용자들은 제일 처음에 리용하였던 파일들에 자주 접근한다.
4. 사용자들은 일반적으로 디스크장치를 직접 열지 못하고 고준위조작체계응용프로그람에 의거한다.
5. 사용자들은 같은 체계에 한번이상 가입할수 없다.
6. 사용자들은 체계프로그람을 복사할수 없다.

아이데스(IDES)에서 리용되는 침입식별도식은 다음과 같은 전략을 대표한다. 검열기록들은 생성되자마자 조사되여 규칙기준과 대조된다. 대조되면 사용자의 의심정도는 증가한다. 대조되는 규칙들이 많으면 의심정도는 극단적인 경우의 한계를 벗어 날것이다.

아이데스방식은 검열기록의 조사에 기초하고 있다. 이 계획의 결함은 융통성이 부족한것이다. 주어 진 침입계획을 약간 변화시키거나 미묘한 방법으로 변화시킬 때마다 여러개의 대리인검열기록렬이 만들어 질수 있다. 아주 명백하게 표현된 규칙들에 의해 이것들의 모든 변화를 반영하는것이 힘들수 있다. 다른 방법은 특정의 검열기록과 독립인 고준위모형을 개발하는것이다. 그러한 실례가 유스타트(USTAT)로서 알려 져 있는 상태이행모형이다(참고문헌 [ILGU93]). 유스타트는 UNIX의 검열기구에 의해 기록된 상세한 특정의 작용보다도 일반적인 작용을 처리한다. 유스타트는 239개의 사건에 대한 검열기록을 제공하는 싼오에쓰(SunOS)체계에서 실현된다. 이것들중 28개만은 전처리기에서 리용되며 이것은 그것들을 10개의 일반적인 작용들로 넘긴다(표 15-4). 바로 이 작용들과 그 매 작용들에서 호출하는 파라메터들을 리용하여 혐의활동을 특징 짓는 상태이행도를 만들수 있다. 검열할수 있는 여러개의 서로 다른 사건들은 보다 작은 여러개의 작용들로 넘어 가므로 규칙-창조처리공정은 매우 간단하다. 더우기 새롭게 알게 된 침입행동들을 나타내기 위해 상태이행도를 변경하는것도 쉽다.

표 15-4. SunOS사건형태에 대한 USTAT작용

USTAT작용	SunOS사건형
Read	open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt
Write	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct
Create	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
Delete	rmdir, unlink
Execute	exec, execve
Exit	exit
Modify_Owner	chown, fchown
Modify_Perm	chmod, fchmod
Rename	rename
Hardlink	link

분산침입검출

지금까지의 침입검출작업은 독립형설비의 단일체계에 주목하였다. 그러나 일반적인 기구에서는 LAN이나 인터네트망이 지원하는 분산된 가입자들의 모임을 방어하는것이 필요하다. 매 가입자에서는 독립형침입검출체계를 리용하여 방어할수 있지만 더 효과적인 방어는 망을 통한 침입검출체계에서 조정과 협조에 의해 달성할수 있다.

포라스(Porras)는 분산침입검출체계의 설계에서 다음과 같은 주요한 결과를 지적하였다[PORR92].

- 분산침입검출체계에서는 다른 검열기록형식을 취하여야 한다. 각이한 환경에서 서로 다른 체계들은 서로 다른 가공하지 않은 검열수집체계를 취할것이며 침입검출을 리용하는 경우 보안과 관련한 검열기록들에 대해 서로 다른 형식을 취할수 있다.
- 망에서 한개 혹은 그이상의 마디들은 망우에서의 체계의 자료에 대한 요점의 수집과 분석으로서 봉사할것이다. 그러므로 가공하지 않은 검열자료나 개요자료는 망을 통하여 전송해야 한다. 따라서 이 자료의 완정성과 기밀성을 보증하여야 할 필요가 제기된다. 완정성은 침입자가 전송한 검열정보를 변경시켜 자기의 활동들을 감추는것을 막기 위해 필요하다. 기밀성은 전송된 검열정보가 변경될수 있으므로 필요하다.
- 집중 혹은 분산방식을 리용할수 있다. 집중방식에는 모든 검열자료를 수집, 분석하는 한개의 중앙점이 있다. 이것은 호상관계가 있는 입력보고서의 과제는 쉽게 처리하지만 강한 《병목》현상과 단일실패점이 나타난다. 분산방식에는 분석중심이 한개이상 있지만 이것들은 자기 활동들을 조정하고 정보를 교환해야 한다.

실례로 데이비스 캘리포니아(Davis California)종합대학에서 개발한 분산검출체계를 들수 있다(참고문헌 [HEBE92, SNAP91]). 그림 15-5에 전체 방식을 보여 주었는데 이것은 3가지 주요구성부분으로 이루어 져 있다.

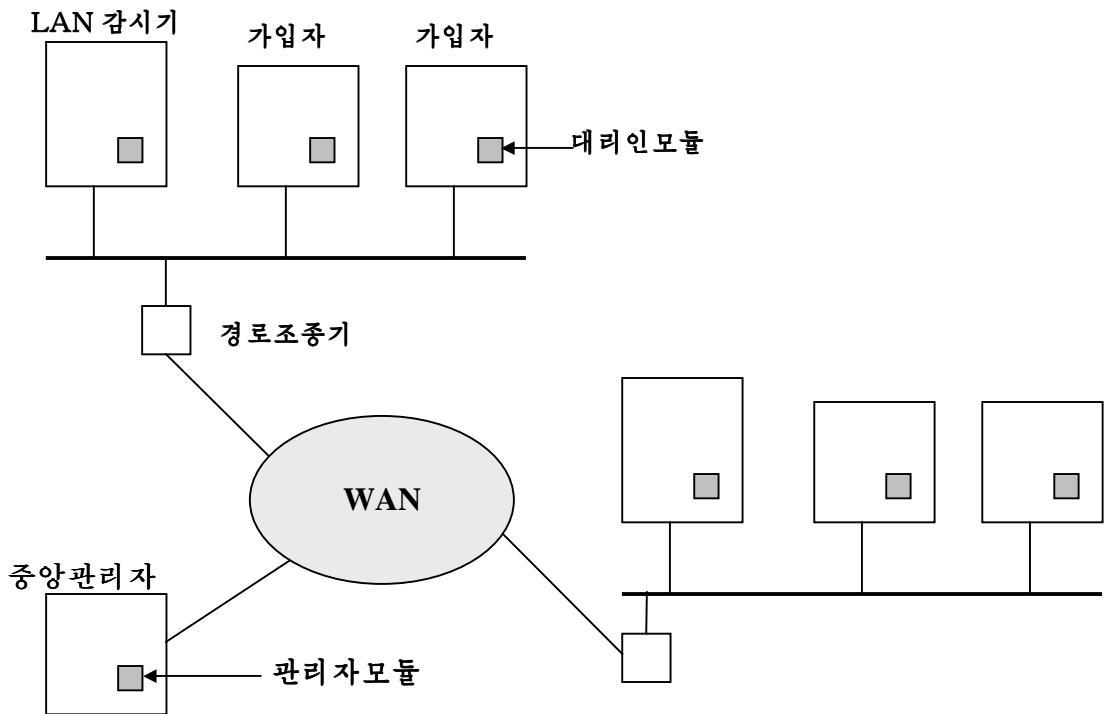



그림 15-5. 분산침입검출방식

- **가입자대리인모듈:**감시체계우에서 배경처리로서 조작하는 검열수집모듈이다. 이것의 목적은 가입자의 보안과 관련한 사건자료들을 수집하여 중앙관리자에게 그 자료들을 전송하는것이다.
- **LAN감시기대리인모듈:** LAN통신을 분석하는것외에 가입자대리인모듈과 같은 방식으로 작용하며 중앙관리자에게 결과를 보고한다.
- **중앙관리자모듈:** LAN감시기와 가입자대리인으로부터 보고를 받아 처리하며 침입을 검출하기 위해 이 보고들을 서로 련관시킨다.

도식은 임의의 조작체계나 검열체계의 실현에 관계되지 않도록 설계한다. 그림 15-6[SNAP91]에 이 도식을 실현하는 일반적인 방식을 보여 주었다.

대리인은 검열수집체계로부터 가공되지 않은 모든 검열기록을 획득한다. 려파기는 보안리익이 있는 기록만을 보존하게 한다.이 기록들은 다음 가입자검열기록(HAR)으로서 표준화된 형식으로 다시 형식화된다. 다음으로 형판구동(template-driven)론리모듈이 혐의활동에 관한 기록들을 분석한다. 제일 낮은 준위에서 대리인은 과거의 임의의 사건들과 독립인 흥미 있는 주목할만 한 사건들에 대하여 세밀히 조사한다. 이러한 사건들의 실례로서는 실패된 파일접근, 체계파일에로의 접근, 파일의 접근조종의 변화를 들수 있다. 다음으로 높은 준위에서 대리인은 알고 있는 공격패턴(서명)과 같은 사건들의 렬을 찾는다. 마지막으로 대리인은 실행된 프로그람의 수, 접근한 파일의 수 등과 같은 사용자들의 리력

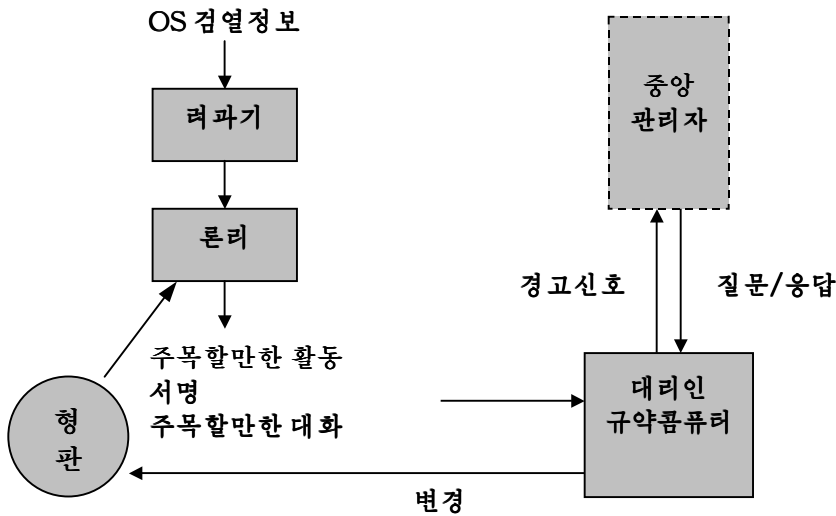



그림 15-6. 대리인방식

륜곽에 기초하고 있는 개별적인 사용자의 례외활동을 찾는다.

협의활동이 검출되면 경고신호를 중앙관리자에게 보낸다. 중앙관리자는 수신된 자료로부터 추론을 진행하는 전문가체계를 포함하고 있다. 관리자는 다른 대리인로부터 HAR들과의 련관성을 나타내기 위해 개별적인 체계에 HAR들의 복사에 대한 질문을 할수도 있다.

LAN감시기의 대리인도 중앙관리자에게 정보를 제공한다. LAN감시기의 대리인은 가입자-가입자접속,리용하는 봉사들, 통신량을 검열한다. 또한 망적재시의 급격한 변화와 같은 의미 있는 사건들, 보안과 관련한 봉사들의 리용, 재가입과 같은 망활동을 찾는다.

그림 15-5와 그림 15-6에서 서술한 방식은 완전히 일반적이고 융통적이다. 그것은 독립형침입검출로부터 체계에로 확장될수 있는 콤퓨터독립형방법의 기초를 제공하는데 그에 의해 많은 싸이트들과 망들로부터 활동을 호상 련관시켜 검출되지 않은채로 있는 협의활동들을 검출할수 있다.

15.2 비루스와 그와 관련된 위협

콤퓨터체계에 대한 가장 세련된 위협형태는 콤퓨터체계의 약점을 리용한 프로그람에 의해 나타난다. 여기서는 응용프로그람과 편집기, 콤파일러, 봉사프로그람을 고찰한다.

먼저 쏘프트웨어의 위협의 범위를 개괄한다. 이 절의 마지막에 비루스에 대해서 론한다. 우선 특성을 보고 대응수단에 대해 고찰한다.

위법프로그람

그림 15-7에 쏘프트웨어에 의한 위협의 총적인 분류와 위법프로그람들을 제시하였다. 이 위협들을 두가지 부류로 나눌수 있다. 가입자프로그람을 요구하는것과 독립적인것이

다. 전자는 일부 실제적인 응용프로그람, 봉사프로그람, 체계프로그람과 독립적으로 존재할수 없는 토막프로그람들이다. 후자는 계획화될수 있으며 조작체계에 의해 실행될수 있는 자립적인 프로그람이다.

자가복제하지 않는 위협쏘프트웨어들과 자가복제하는 위협쏘프트웨어로 구별할수도 있다. 먼저의것은 가입자프로그람이 특정한 함수를 실행하려고 접근할 때 동작하는 파괴프로그람이다. 그다음의것은 실행할 때 같은 체계나 어떤 다른 체계우에서 후에 동작하여야 할 한개 혹은 그이상의 자체복사들을 만들수 있는 파괴프로그람(비루스)이나 독립적인 프로그람(웜, 박테리아)으로 이루어 진다.

비록 그림 15-7의 분류는 론의하는 정보를 종합하는데 가치는 있으나 모든 분류가 다 가치 있는것은 아니다. 특히 론리폭탄이나 트로이목마는 비루스나 웜의 일부분으로 될수 있다.

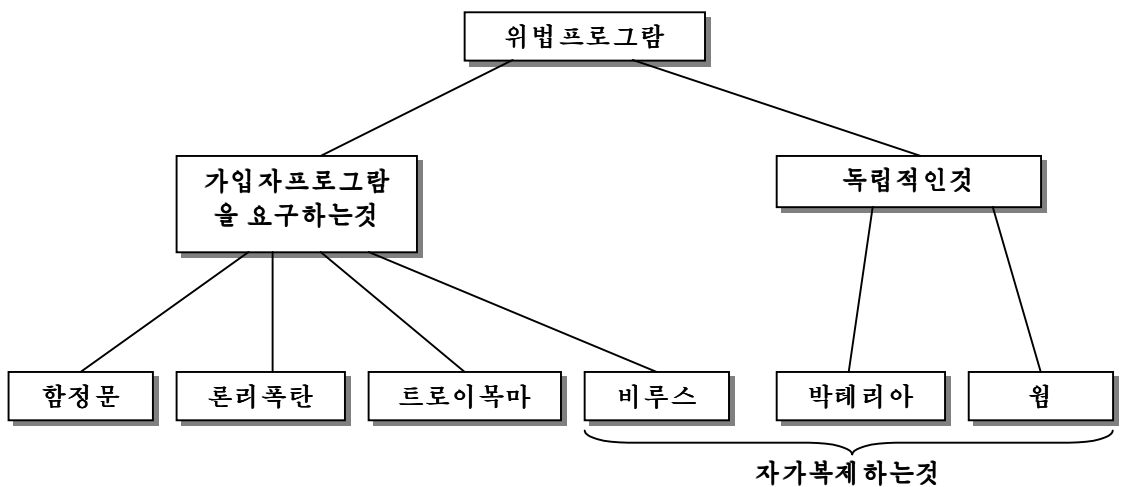



그림 15-7. 위법프로그람의 분류

함정문

함정문은 함정문을 아는 그 누군가가 보안접근수속을 거치지 않고 접근하게 하는 프로그람에 대한 비밀입력자료점이다. 함정문은 프로그람작성자들이 프로그람을 오유수정하는데 다년간 합법적으로 리용되여 왔다. 함정문은 프로그람작성자가 인증수속을 가지거나 사용자가 응용프로그람을 실행하기 위해 많은 다른 값들을 입력하는데 오랜 설정조작이 요구될 때 설치된다. 프로그람을 오유수정하기 위해 개발자는 특별한 특권준위를 얻으려고 하거나 모든 필요한 설정과 인증을 피하려고 할수 있다. 프로그람작성자는 응용프로그람에 만들어 진 인증수속으로써 무엇인가 틀리게 프로그람을 동작시키는 방법이 있다는것을 보증하려고도 할수 있다. 함정문은 어떤 특별한 입력자료렬을 평가하는 부호이거나 정확한 사용자의 식별자에 의해 혹은 적합하지 않은 사건렬에 의해 발화되는 부호이다.

함정문들은 위법프로그람작성자들이 부여되지 않은 접근자격을 얻으려고 이것들을 리용할 때 위협으로 된다. 함정문의 기본사상은 《전쟁오락(*War Games*)》(참고문헌

[COOP89])영화에서 묘사한 약점을 리용하였다는것이다. 또 다른 실례는 말틱스(Multics)의 개발기간 공군(Air Force)의 범팀(적수들을 모의하여)이 침입검사들을 진행하였다는것이다. 채용한 한가지 전략은 말틱스를 실행하는 싸이트에 갱신된 위조조작체계를 보내는것이다. 갱신은 함정문으로서 동작하여 범팀이 접근자격을 얻게 하는 트로이목마를 포함하였다. 그후에 위협의 존재를 알았을 때에도 말틱스의 개발자들은 그것을 찾지 못했으므로 위협은 훌륭히 실현되였다[ENGE80].

함정문이 조작체계를 조종하게 하는것은 힘들다. 보안측정수단들은 프로그람개발과 쏘프트웨어의 갱신활동에 주목하여야 한다.

론리폭탄

비루스나 웜보다 먼저 나온 가장 오랜 형태의 하나가 론리폭탄이다. 론리폭탄은 일정한 조건을 만날 때 《폭발》하는 어떤 합법적인 프로그람에 매몰된 부호이다. 론리폭탄이 리용할수 있는 발화조건의 실례로서는 일정한 파일의 존재나 결여, 한주의 특별한 날자나 요일, 응용프로그람을 실행하는 특별한 사용자 등이다. 한가지 유명한 경우로서(참고문헌 [SPAF89]) 론리폭탄은 어떤 사용자에 대한 식별자번호를 검사한 다음 두개의 련속적인 종업원명부처리에서 나타나지 않으면 발화된다. 일단 발화되면 폭탄은 자료나 파일전체를 변경시키거나 삭제하며 콤퓨터의 폭주를 일으키는 등의 손상을 준다. 론리폭탄을 어떻게 리용할수 있는가에 대한 인상적인 실례는 메리랜드주의 몬트고메리시의 도서관체계(Montgomery County, Maryland, library)(참고문헌 [TIME90])의 경우를 들수 있다. 콤퓨터화된 류통체계를 개발한 계약인은 돈을 지불하지 못하면 일정한 날자에 체계를 쓸수 없게 하는 론리폭탄을 삽입하였다. 체계가 불충분한 응답시간을 가졌으므로 도서관이 그것의 최종적인 지불을 지연하였을 때 계약인은 폭탄의 존재를 밝히고 곧 지불하지 않으면 폭발하게 될것이라고 위협하였다.

트로이목마

트로이목마는 요구될 때 필요 없거나 해로운 기능을 수행하는 숨은 부호가 들어 있는 쓸모 있거나 쓸모 있어 보이는 프로그람이나 지령수속이다.

트로이목마프로그람을 리용하여 권한이 없는 사용자가 직접 얻을수 없는 기능을 간접적으로 얻을수 있다. 실례를 들어 공유한 체계우에서 다른 사용자의 파일에 접근하기 위해 사용자들은 임의의 사용자가 파일을 읽을수 있게 호출되는 사용자파일의 허용을 변화시키는 트로이목마프로그람을 만들수 있다. 다음 공동등록부에 그것을 배치하고 유용한 응용프로그람으로 나타나도록 이름을 붙임으로써 사용자가 프로그람을 실행하게 한다. 실례로 자신이 바라는 형태로 사용자의 파일들을 직관적으로 렬거하는 프로그람을 들수 있다. 다른 사용자가 이 프로그람을 실행할 때 사용자파일에 대한 정보를 호출할수 있다. 검출하기 힘든 트로이목마프로그람의 실례는 체계가입프로그람과 같은 일정한 프로그람들에 이것들을 콤파일할 때 추가적인 부호를 삽입하도록 변경시킨 콤파일러이다[THOM84]. 부호는 개발자가 특수한 통과암호를 리용하여 체계에 가입하게 하도록 가입프로그람에 함정문을 만든다. 이 트로이목마는 가입프로그람의 원천부호를 읽는것으로써는 결코 발견할수 없다.

트로이목마는 또한 일반적으로 자료를 파괴한다. 그 프로그람은 유용한 기능을 실행하는것처럼 나타나지만(즉 전자수판프로그람) 사용자의 파일을 조용히 삭제할수도 있다. 례를 들어 CBS집행위원회는 콤퓨터기억기에 기억된 모든 정보를 파괴하는 트로이목마에 의해 피해를 입었다[TIME90]. 그때 트로이목마는 전자게시판체계에서 제출한 그라픽스루틴에 끼워 져 있었다.

비루스

비루스는 자기를 변경하여 다른 프로그람을 전염시킬수 있는 프로그람이다. 변경은 비루스프로그람의 복사를 포함하며 이것은 계속하여 다른 프로그람에 전염되여 간다.

생물비루스는 살아 있는 세포의 구조체에 들어 붙어 본래의 비루스를 수많이 완전히 복제하는 기교를 물려 줄수 있는 유전자부호 DNA나 RNA의 작은 부분이다. 콤퓨터비루스는 완전한 자가복제를 만드는 부호를 자기의 명령부호에 가지고 있다. 일반적으로 비루스는 가입자콤퓨터에 잠복하며 콤퓨터디스크조작체계의 일시적인 조종을 리용한다. 다음 전염된 콤퓨터가 전염되지 않은 쏘프트웨어의 부분과 접촉할 때 항상 새로운 프로그람에 비루스의 새로운 복사가 진행된다. 그러므로 디스크를 교환하거나 망우에서 어떤 다른 곳으로 프로그람을 보내는 사용자에 의해 콤퓨터에서 콤퓨터에로 전염될수 있다. 망환경에서는 다른 콤퓨터에 있는 응용프로그람과 체계봉사를 호출할수 있으므로 비루스가 전파될수 있다.

비루스들에 대해서는 이 절의 뒤부분에서 구체적으로 고찰한다.

웜

망의 웜프로그람들은 망접속을 리용하여 체계에서 체계에로 전파된다. 일단 체계내에 존재하면 웜은 콤퓨터비루스나 박테리아로서 행동할수 있으며 트로이목마프로그람들을 끼워 넣거나 혹은 여러가지 중단이나 파괴동작을 할수 있다.

자기자신을 복사하기 위하여 망웜은 대체로 어떤 망운반수단을 리용한다. 실례로 다음과 같은것을 들수 있다.

- **전자우편기능:** 웜은 다른 체계에 자신의 복사를 우편으로 보낸다.
- **원격실행능력:** 웜은 다른 체계에서 자신의 복사를 진행한다.
- **원격가입능력:** 웜은 사용자처럼 먼 곳에 있는 체계에 가입한 다음 한 체계로부터 다른 체계에로 자기자신을 복사하는 지령을 실행한다.

웜프로그람의 새로운 복사는 먼 곳에 있는 체계에서 실행하는 한편 이 체계에서 수행하는 임의의 기능외에 같은 방법으로 계속 전파한다.

망웜은 콤퓨터비루스와 같은 특징을 가진다. 즉 잠복단계, 전염단계, 발병단계, 실행단계를 가진다. 전염단계는 일반적으로 다음의 수속에 따라 진행된다.

1. 가입자표나 먼 곳에 있는 체계의 주소와 류사한 저장고를 조사하여 전염하려는 어떤 체계를 찾는다.
2. 먼 곳에 있는 체계와 접속한다.
3. 먼 곳에 있는 체계에 자신을 복사하고 복사가 실행되게 한다.

웜은 체계에 자기자신을 복사하기전에 먼저 전염시키려는 체계가 있는 곳을 찾는다. 다중프로그람작성체계에서는 체계조작자가 알아 차릴수 없게 체계처리공정으로서 자신의 이름을 만들거나 어떤 다른 이름을 리용하여 잠복할수 있다.

비루스와 같이 웜도 예방하기가 힘들다. 그러나 망보안과 단일체계보안수단을 정확히 설계하고 실현하면 웜의 위협을 최소화할수 있다.

박테리아

박테리아는 명백히 아무 파일이나 파괴하지 않는 프로그람이다. 그것의 유일한 목적은 자기자신을 복제하는것이다. 일반적으로 박테리아프로그람은 다중프로그람작성체계에

서 동시에 두개의 자가복제를 진행하거나 두개의 새로운 파일을 창조하는것외에는 아무것도 하지 않으며 그 매개 파일은 박테리아프로그람의 본래의 원천파일의 복사이다. 다음 이 두 프로그람은 자기자신을 두배로 복사할수 있다. 박테리아는 지수함수적으로 복제되며 언젠가는 처리기용량, 기억기, 디스크공간을 전부 차지하여 이 자원들에 대한 사용자의 접근을 배제한다.

비루스의 본질

비루스는 다른 프로그람이 하는 모든것을 할수 있다. 다만 차이는 다른 프로그람에 자기자신을 첨부하며 숙주프로그람이 실행될 때 비밀리에 실행된다는것이다. 일단 비루스가 실행되면 그것은 파일과 프로그람을 지우는것과 같은 기능을 수행한다. 일반적으로 비루스는 다음의 4가지 단계를 거친다.

- **잠복단계:** 비루스는 아무것도 하지 않는다. 비루스는 어떤 날자, 다른 프로그람이나 파일의 존재, 일정한 한계를 벗어 나는 디스크의 용량과 같은 어떤 사건에 의해 마침내 동작할수 있을것이다. 모든 비루스가 다 이 단계를 거치는것은 아니다.
- **전염단계:** 비루스는 디스크에 있는 다른 프로그람이나 어떤 체계에 자가복제를 진행한다. 전염된 매 프로그람은 그때부터 비루스클론을 포함하고 있게 되며 이것은 자체로 전염단계에 들어 갈수 있다.
- **발병단계:** 비루스는 자기가 하려고 하는 기능을 수행하기 위해 동작한다. 잠복단계와 같이 발병단계는 비루스가 몇번 복사되였는가 하는 회수의 계수를 비롯하여 각이한 체계의 사건에 의해 일어 날수 있다.
- **실행단계:** 기능을 수행한다. 기능의 실행은 화면에 통보문을 내보내는것과 같이 해를 주지 않거나 프로그람과 자료파일의 파괴와 같은 피해를 줄수 있다.

대부분의 비루스들은 조작체계에 따라 특별하거나 하드웨어의 가동환경에 따라 특별한 의미에서 작업을 수행한다. 그러므로 그것들은 매개 체계의 구체적인 내용과 약점을 리용하여 설계되였다.

비루스의 구조

비루스는 실행가능한 프로그람에 미리 전염되거나 후에 전염될수 있으며 혹은 어떤 다른 방식으로 들어 있을수 있다. 비루스조작의 기본은 전염된 프로그람이 실행될 때 먼저 비루스부호를 실행하고 다음 프로그람의 원천부호를 실행하게 한다는것이다.

가장 일반적인 비루스의 구조에 대한 표상을 그림 15-8에 보여 주었다[COHE94]. 이 경우에 비루스부호 V는 전염된 프로그람에 미리 잠복해 있으며 실행되는 프로그람의 입력자료점은 프로그람의 첫행이라고 가정한다.

전염된 프로그람은 비루스부호부터 시작하여 다음과 같이 실행된다. 부호의 첫행은 기본비루스프로그람에로 이행하는것이다. 두번째 행은 이미 비루스에 전염된 프로그람인가 아닌가를 결정하기 위해 비루스가 리용하는 전문표식자이다. 프로그람이 호출될 때 조종은 즉시 기본비루스프로그람에로 이행한다. 비루스프로그람은 처음 전염되지 않은 실행가능한 파일을 찾아 보고 그것을 전염시킨다. 다음 비루스는 어떤 작용을 수행하며 보통 체계에 손상을 준다. 이 작용은 프로그람이 호출될 때마다 매번 수행될수 있거나 일정한 조건하에서만 발화되는 론리폭탄으로 될수 있다. 마지막으로 비루스는 본래 프로

그람에로 조종을 이행한다. 프로그람의 전염단계가 매우 빠르면 사용자는 전염된 프로그람과 전염되지 않은 프로그람사이의 임의의 차이점을 알아 차리지 못한다.

```
        Program V:=

    {goto main:
        1234567:

          subroutine infect-executable:=
              {loop:
            file:=get-random-executable-file;
            if(first-line-of-file=1234567)
                    then goto loop
                    else prepend V to file;}

               subroutine do-damage:=
              {whatever damage is to be done}

               subroutine trigger-pulled:=
              {return true if some condition holds}

    main:     main-program:=
              {infect-executable;
              if trigger-pulled then do-damage;
              goto next;}

    next:

    }
```

그림 15-8. 간단한 비루스

우에서 서술한것과 같은 비루스는 전염된 프로그람이 대응하는 전염되지 않은 프로그람보다 커지기때문에 쉽게 검출된다. 비루스를 검출하는 이와 같은 간단한 방법을 막는 방도는 전염된것과 전염되지 않은 프로그람이 같은 크기를 가지도록 실행가능한 파일을 압축하는것이다. 그림 15-9(참고문헌 [COHE94])에 일반적으로 필요한 론리를 보여주었다. 이 비루스에서 기본행은 번호로 되여 있으며 그림 15-10(참고문헌 [COHE94])에 조작을 보여 주었다. 프로그람 P_1는 비루스 CV에 전염되여 있다. 이 프로그람이 호출될 때 조종은 비루스에로 넘어 가며 다음과 같은 단계를 수행한다.

```
        program CV:=

{goto main:
          01234567;

        subroutine infect-executable:=
                {loop:
                        file:=get-random-executable-file;
                if(first-line-of-file = 01234567) then goto loop;
          (1)   compress file;
          (2)   prepend CV to file;
                }

main: main-program :=
                {if ask-permission then infect-executable;
          (3)   uncompress rest-of-file;
          (4)   run uncompressed file;}
          }
```

그림 15-9. 압축비루스의 론리

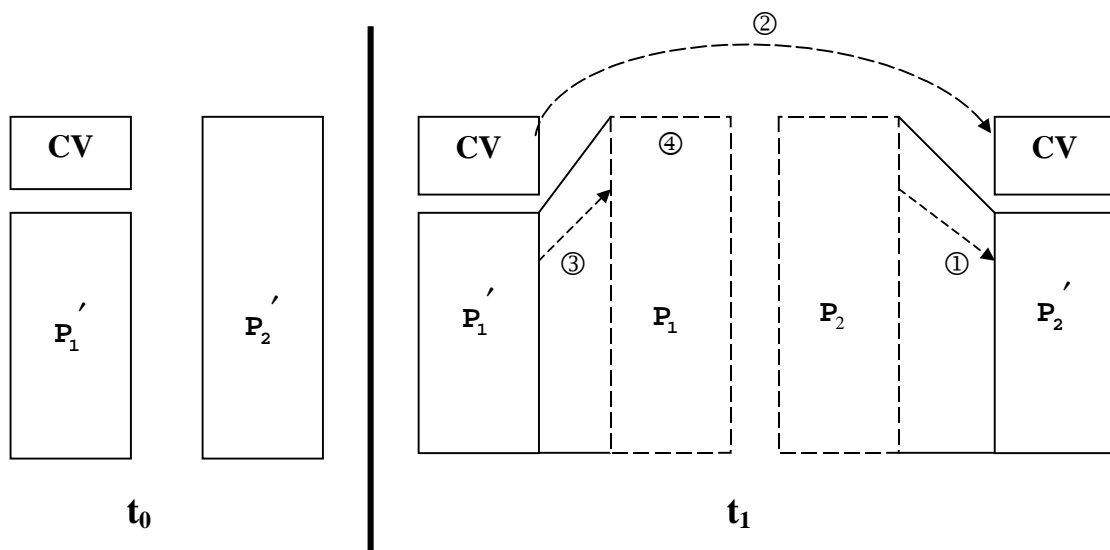



그림 15-10. 압축비루스

1. 비루스는 처음에 찾아 낸 개개의 전염되지 않은 파일 P_2을 본래의 프로그람보다 비루스의 크기만큼 더 작게 P_1'로 압축한다.
2. 압축된 프로그람에 비루스가 복사된다.
3. 본래 전염된 프로그람의 압축된 판본 P_1'를 푼다.
4. 압축을 푼 본래의 프로그람이 실행된다.

이 실례에서 비루스는 전염외에 아무것도 하지 않는다. 이전의 실례에서처럼 비루스는 론리폭탄을 포함할수 있다.

초기전염

일단 비루스가 한개의 프로그람에 전염되여 체계에 대한 가입을 얻으면 그것은 전염된 프로그람이 실행될 때 이 체계의 일부 혹은 모든 다른 실행가능한 파일에 전염될수 있는 상태에 있다. 그러므로 첫번째 장소에 대한 가입을 얻는것을 막음으로써 전염을 완전히 방지할수 있다. 유감스럽게도 비루스는 체계를 제외하고는 그 어떤 프로그람의 부분으로도 될수 있으므로 방지는 매우 힘들다. 그러므로 깨끗이 청소한 다음 체계와 응용프로그람을 쓰지 않는 한 위험하다.

비루스들의 전염은 대부분 디스크에 있는 프로그람들을 콤퓨터에 복사할 때 일어 난다. 이러한 디스크들로서는 대부분 오락이 들어 있는 디스크나 사용자들이 가정용콤퓨터용으로 구입하여 사무콤퓨터에 넣어 쓰는 간단하고 쓰기 편리한 응용프로그람들이 있는 디스크들이다. 믿기 어렵지만 어떤것은 응용프로그람제작자가 포장한 디스크들에도 존재한다. 다만 전염의 일부만이 망접촉을 통해 진행된다. 이것들중 대부분은 전자게시판체계로부터 구한다. 일반적으로 사용자는 비루스가 들어 있다는것을 알아 차리기전까지는 오락이나 가치 있는 응용프로그람들을 계속 끌어 올것이다.

비루스의 형태

처음 비루스가 출현한 때로부터 비루스제작자와 항비루스쏘프트웨어제작자들사이의 경쟁은 끊임없이 계속되고 있다. 이미 있는 비루스형태에 대해서 효과적인 수단들이 개발되여 온것처럼 새로운 비루스형태들도 개발되고 있다. 참고문헌 [STEP93]에서는 가장 중요한 비루스형태로서 다음과 같은 부류를 제시하였다.

- **기생비루스:** 전통적이며 여전히 일반적인 대부분의 비루스형태이다. 기생비루스는 실행가능한 파일에 붙어서 전염된 프로그람이 실행될 때 다른 실행가능한 파일을 찾아서 복제한다.
- **기억기상주형비루스:** 상주체계프로그람의 부분으로서 주기억기에 들어 가 있다. 이 순간부터 비루스는 실행하는 모든 프로그람에 전염된다.
- **기동분구비루스:** 주기동기록이나 기동기록에 전염되며 체계가 비루스가 있는 디스크로부터 기동될 때 전파된다.
- **스텔스비루스:** 항비루스쏘프트웨어의 검출로부터 자기자신을 완전히 감추도록 설계된 비루스형태이다.

- **다형성비루스:** 비루스의 《서명》에 의한 검출이 불가능하며 매번 전염되는것과 동시에 변화되는 비루스이다.

스텔스비루스의 한가지 실례는 다음과 같다. 압축을 리용하여 전염된 프로그람이 전염되지 않은 프로그람과 꼭같은 길이를 가지게 하는 비루스이다. 훨씬 더 정교한 기술이 있을수 있다. 례를 들어 비루스는 디스크 I/O루틴에 대한 론리를 가로 챌수 있으면 이 루틴을 리용하여 사용자가 의심스러운 부분을 읽으려고 할 때 반대로 본래의 프로그람 즉 전염되지 않은 프로그람이 나타나게 한다. 이렇게 비밀은 비루스에 적용되는것이 아니라 반대로 비루스가 검출을 피하는데 리용되는 기술이다.

다형성비루스는 기능적으로는 동등하지만 복제할 때에 명백히 차이나는 비트패턴을 가지는 복사를 창조한다. 비밀비루스와 같이 목적은 비루스를 조사하는 프로그람을 좌절시키는것이다. 이 경우에는 복사할 때마다 비루스의 《서명》을 매번 변화시킨다. 서명을 변화시키기 위하여 비루스는 쓸데 없는 명령을 무질서하게 삽입하거나 개별적인 명령들의 순서를 교환한다. 가장 효과적인 방식은 암호를 리용하는것이다. 일반적으로 **변이엔진**이라고 부르는 비루스의 일부분은 비루스의 남은 부분을 암호화하기 위하여 란수적인 암호열쇠를 만든다. 열쇠는 비루스와 함께 기억되며 변이엔진은 자체로 변경된다. 전염된 프로그람이 호출될 때 비루스는 자기를 복호하기 위해 기억된 우연적인 열쇠를 리용한다. 비루스는 복제될 때 다른 우연수열쇠를 선택한다.

비루스제작자의 《무기고》에 있는 또 다른 비루스무기는 비루스를 만드는 도구묶음이다. 이런 도구묶음은 《풋내기》도 다른 여러가지 비루스들을 빨리 만들수 있게 한다. 비록 도구묶음으로서는 완전히 빈터에서 설계한 비루스들보다 좀 더 복잡하게 비루스들을 만들수 있지만 생성될수 있는 여러가지 완전히 새로운 비루스들은 항비루스전략에 대한 문제를 만들어 낸다. 또 다른 비루스제작도구는 비루스교환게시판이다. 미국과 기타 나라들에서 이와 같은 게시판들이 급속히 늘어 나고 있다(참고문헌 [ADAM92]). 이 게시판들은 비루스를 만드는 비결뿐아니라 비루스들의 복사물들을 제공한다.

마크로비루스

최근 많은 비루스들이 회사싸이트들과 충돌하는 회수가 급격히 늘어 났다(참고문헌 [BERG97]). 사실 이 충돌은 보다 새로운 형태의 비루스들의 확산때문이였다. 국가콤퓨터보안기관(www. nsca. com)에 의하면 마크로비루스는 지금 모든 콤퓨터비루스들의 2~3배로 만들어 지고 있다.

마크로비루스는 여러가지 리유로 특히 위험하다.

1. 마크로비루스는 가동환경과는 독립이다. 사실 모든 마크로비루스들은 MicrosoftWord문서에 전염된다. 워드를 지원하는 임의의 하드웨어가동환경과 조작체계에 전염될수 있다.
2. 마크로비루스는 실행가능한 부호부분이 아니라 문서에 전염된다. 콤퓨터체계에 들어 있는 대부분의 정보들은 프로그람보다도 문서형태로 되여 있다.
3. 마크로비루스들은 매우 쉽게 전파된다. 가장 일반적인 방법은 전자우편에 의

거하는것이다.

마크로비루스들은 워드나 MicrosoftExel과 같은 다른 사무처리응용프로그람에 있는 특징들을 리용한다(이름그대로 마크로이다). 본질에 있어서 마크로는 워드가 처리하는 문서나 다른 파일형태에 매몰된 실행가능한 프로그람이다. 일반적으로 사용자들은 과제를 자동적으로 반복수행하기 위해 마크로를 리용하며 이것에 건누름동작을 기억시킨다. 마크로언어는 보통 베이지크프로그람작성언어의 한 형식이다. 사용자는 마크로에 건누름렬을 정의하고 기능건이나 전문적인 짧은 건결합을 입력할 때 마크로가 호출되도록 설정할수 있다. 보통 자동실행가능한 사건들은 파일열기, 파일닫기, 응용프로그람실행이다. 일단 마크로가 실행되면 비루스는 다른 문서에 자기자신을 복제하며 파일을 지우고 사용자의 체계에 여러가지 다른 피해를 줄수 있다. MicrosoftWord에는 자동실행가능한 마크로로서 3가지 형태가 있다.

- **자동실행:** 자동실행이라고 부르는 마크로가 《normal. dot》형타나 워드를 설치한 등록부에 기억된 대역형타에 있으면 워드가 시작할 때마다 항상 실행된다.
- **자동마크로:** 문서를 열거나 닫고 새로운 문서를 만들거나 워드를 끌내는것과 같은 정의된 사건이 발생할 때마다 자동마크로가 실행된다.
- **지령마크로:** 대역마크로파일에 있는 마크로나 문서에 붙어 있는 마크로가 워드지령의 이름을 가질 때 지령마크로는 사용자가 지령(즉 File Save)을 호출할 때마다 항상 실행된다.

마크로비루스가 전파되는 일반적인 방법은 다음과 같다. 자동마크로나 지령마크로는 전자우편이나 디스크를 전송함으로써 체계에 들어 있는 워드문서에 전파된다. 문서를 연 다음 어떤 순간에 마크로가 실행된다. 마크로는 대역마크로파일에 자기자신을 복사한다. 워드의 다음 대화가 열릴 때 전염된 대역마크로가 동작한다. 이 마크로는 실행될 때 자가복제하며 피해를 준다.

련속적인 워드의 실현들에서 마크로비루스에 대해 강한 보호가 제공되고 있다. 례를 들어 Microsoft회사는 의심스러운 워드파일을 검출하고 마크로로써 파일을 열 때 가능한 위험에 대해 사용자에게 경고하는 만능마크로비루스보호도구를 제공한다. 여러 분야의 항비루스제품판매자는 마크로비루스를 검출하고 정정하는 도구들도 개발하고 있다. 다른 형태의 비루스와 마찬가지로 마크로비루스분야에서도 경쟁이 계속되고 있다.

항비루스방식

비루스의 위협을 리상적으로 해결하는 방법은 예방하는것이다. 즉 비루스가 처음에 체계에 들어 오지 못하게 하는것이다. 예방이 비루스의 성공하는 공격수를 줄일수는 있어도 보안목표를 달성하는것은 불가능하다. 더 좋은 방식은 다음과 같다.

- **검출:** 일단 전염이 일어 나면 비루스가 발생하였다고 보고 그것을 찾아 낸다.
- **식별:** 일단 검출이 진행되면 전염된 파일이 가지고 있는 특정한 비루스가 식별

된다.

- **제거:** 일단 특정한 비루스가 식별되면 전염된 파일로부터 모든 비루스들을 추적하여 제거하고 그것을 본래의 상태로 회복한다. 전염된 모든 체계로부터 비루스를 제거함으로써 병이 계속 전파될수 없게 한다.

검출은 계속되지만 식별이나 제거를 할수 없으면 전염된 프로그람을 버리고 그대신 깨끗한 예비판을 다시 넣는다.

비루스기술과 항비루스기술의 개선은 밀접한 관계에 있다. 초기의 비루스는 비교적 단순히 파괴만 하는 부호였으므로 단순한 항비루스쏘프트웨어제품으로써 식별하고 제거할수 있었다. 비루스와 함께 《비루스무기경쟁》이 점차 발전할수록 항비루스쏘프트웨어도 필연적으로 더 종합적이고 복잡하게 되였다.

참고문헌 [STEP93]에서는 항비루스쏘프트웨어의 4가지 생성을 지적하였다.

- 1세대:간단한 스캐너
- 2세대:발견적스캐너
- 3세대:동작전략
- 4세대:완전한 보호

1세대스캐너는 비루스를 식별하기 위해 비루스서명을 요구한다. 비루스는 《윌드카드》를 포함할수 있지만 본질상 모든 복사에서 같은 구조와 비트패턴을 가진다. 이와 같은 서명-특정스캐너는 알고 있는 비루스만 검출한다. 1세대스캐너의 다른 형태는 프로그람의 크기기록을 보존하고 크기변화를 찾는것이다.

2세대스캐너는 특정한 서명에 의거하지 않는다. 오히려 예상할수 있는 비루스전염을 찾는 탐색규칙을 리용한다. 이런 스캐너의 한가지 클라스는 흔히 비루스와 관련된 부호의 파괴를 찾는다. 실례를 들어 스캐너는 다형성비루스에서 리용하는 암호순환의 시작을 찾고 암호열쇠를 회복할수 있다. 일단 열쇠가 회복되면 스캐너는 비루스를 식별하기 위해 그것을 복호할수 있으며 다음 그것을 제거하고 봉사프로그람에로 돌아 간다.

다른 2세대방식은 완정성검사이다. 검사합을 매 프로그람에 첨가할수 있다. 비루스가 프로그람에 전염될 때 검사합을 변화시키는것이 매우 복잡하도록 비루스를 막기 위한 암호화된 하쉬함수를 리용할수 있다. 암호열쇠는 비루스가 새로운 하쉬부호를 생성하고 그것을 암호화할수 없게 프로그람마다 따로따로 기억시킨다. 단순한 검사합보다도 하쉬함수를 리용함으로써 비루스가 이전과 같은 하쉬부호를 만들기 위해 프로그람을 조정하는것을 막는다.

3세대프로그람은 전염된 프로그람의 구조보다도 오히려 그의 작용에 의해 비루스를 식별하는 기억기상주프로그람이다. 이와 같은 프로그람은 넓은 비루스령역에 대한 서명과 탐색수법을 개발하는것이 필요하지 않다는 우점을 가진다. 오히려 그것은 전염동작을 가리키는 작은 작용들의 모임을 식별한 다음 그것들을 조정하는것만이 필요하다.

4세대제품은 접속할 때 리용되는 각이한 종류의 항비루스기술을 포함한 프로그람들이다. 이것은 조사하는것과 활동전략으로 구성되여 있다. 또한 이와 같은 프로그람은 체

계에 침입하는 비루스의 능력과 전염시키기 위해 파일들을 갱신하는 비루스의 능력을 제한하는 접근조종능력을 가지고 있다.

《무기경쟁》은 계속되고 있다. 4세대제품과 함께 더 일반적인 목적의 콤퓨터보안대응수단에 의한 방어공간이 넓어 지면서 더 종합적인 방어전략을 리용할것이다.

개선된 항비루스기술

더 세련된 항비루스방식과 제품이 계속 출현하고 있다. 이 절에서는 두가지 가장 중요한것을 강조한다.

일반적인 복호

일반적인 복호(GD)기술은 조사속도가 빠르면서도 매우 복잡한 다형성비루스까지도 쉽게 검출할수 있는 항비루스프로그람을 만들수 있게 한다(참고문헌 [NACH97]). 다형성비루스에 전염된 파일이 실행될 때 비루스는 자기자신을 동작하기 위해 복호하여야 한다. 이런 구조를 검출하기 위해 실행가능한 파일들을 GD스캐너를 통과하여 실행시키는데 GD스캐너는 다음의 요소들로 이루어 진다.

- **CPU모의기:** 쏘프트웨어에 기초한 가상콤퓨터이다. 실행가능한 파일에 있는 명령들은 하위처리기에서 실행되지 않고 모의기에 의해 해석된다. 모의기는 하위처리기가 모의기에서 해석된 프로그람에 의해 영향을 받지 않도록 모든 등록부들과 하드웨어처리기의 쏘프트웨어판본을 포함한다.
- **비루스서명스캐너:** 알고 있는 비루스서명을 찾기 위해 목적부호를 조사하는 모듈이다.
- **모의기조종모듈:** 목적부호의 실행을 조종한다.

매 모의가 시작될 때 모의기는 목적부호의 명령을 해석하기 시작한다. 따라서 부호가 복호화하는 복호루틴을 포함하고 있으면 그것이 해석되는 순간부터 비루스가 폭로되게 된다. 조종모듈은 주기적으로 비루스서명에 대한 목적부호를 주사하기 위해 해석을 중단한다.

해석하는 동안은 목적부호가 완전히 조종되는 환경에서 해석되므로 그것은 현재의 개인용콤퓨터환경에 피해를 주지 않을수 있다.

GD스캐너설계의 가장 힘든 문제점은 매 해석이 얼마나 오래동안 실행되는가를 결정하는것이다. 일반적으로 비루스의 요소들은 프로그람이 실행을 시작하자마자 곧 활성화되지만 이것은 문제로 되지 않는다. 보다 풍부한 스캐너는 개별적인 프로그람을 모의하고 임의의 숨은 비루스를 발견한다. 그러나 항비루스프로그람은 제한된 시간과 자원의 량만을 취하므로 사용자들의 요구를 충족시키지 못한다.

수자면역체계

수자면역체계는 IBM이 개발한 비루스보호에 대한 종합적인 방식이다(참고문헌 [KEPH97a, KEPH97b]). 이 개발의 동기는 인터네트에 기초한 비루스전파의 위협이 심각해 진데 있다. 먼저 이 위협에 대해 간단히 고찰하고 IBM의 방식을 개괄한다.

현재 비루스위협은 새로운 비루스와 새로운 변종의 비교적 느린 전파로 특징 지어진다. 항비루스쏘프트웨어는 일반적으로 월을 단위로 갱신되며 이것은 비루스문제를 조종하는데는 충분하였다. 현재는 인터네트를 통해서 비루스가 비교적 적게 전파된다. 그러나 참고문헌 [CHES97]에서 지적한바와 같이 인터네트기술에서의 두가지 주요한 추세는 앞으로 비루스전파속도에 주는 영향을 증가시킬것이다.

- **통합전자우편체계:**Lotus Notes, Microsoft Outlook와 같은 체계들에서는 그 누구에게 임의의것을 보내거나 받은 객체로써 작업하는것이 매우 단순하다.
- **이동성프로그람체계:**Java나 ActiveX와 같은 체계들은 한 체계에서 다른 체계에로 프로그람들을 옮기게 한다.

이러한 인터네트의 능력에 의해 생겨 난 위협에 대응하여 IBM은 원형수자면역체계 (Prototype Digital Immune System)를 개발하였다. 이 체계는 앞절에서 론의한 프로그람모의기리용의 확장이며 일반목적모의기와 비루스검출체계를 제공한다. 이 체계의 목표는 비루스들이 들어 오자마자 즉시 거의 제거할수 있도록 신속한 응답시간을 제공하는것이다. 새로운 비루스가 기관에 들어 오면 면역체계는 자동적으로 비루스를 포착, 분석, 검출, 차페, 제거하며 다른 곳에서 실행되기전에 비루스를 검출할수 있도록 IBM AntiVirus를 실행하는 체계에 이 비루스에 대하여 알려 준다.

그림 15-11에 수자면역체계조작에서의 전형적인 단계를 보여 주었다.

1. 개개의 PC에 있는 감시조종프로그람은 체계의 행동, 의심스러운 프로그람의 변화, 비루스가 나타날수 있다고 암시하는 서명계렬에 기초한 여러가지 발견적탐색법을 리용한다. 감시조종프로그람은 기구내에 있는 관리기에 전염되였다고 생각하는 임의의 프로그람의 복사를 발송한다.
2. 관리기는 표본을 암호화하고 중앙비루스해석기에 그것을 보낸다.
3. 이 콤퓨터는 전염된 프로그람을 안전하게 해석할수 있는 환경을 창조한다. 이 목적에 리용되는 기술들은 수상한 프로그람을 실행하고 조종할수 있는 보호된 환경의 모의나 창조를 포함한다. 다음 비루스해석기는 비루스를 식별하고 제거하기 위한 지령을 만든다.
4. 지령의 결과를 관리기에 되돌려 보낸다.
5. 관리기는 전염된 의뢰기에로 지령을 발송한다.
6. 기구내에 있는 다른 의뢰기에도 지령을 발송한다.
7. 인터네트의 모든 가입자들은 새로운 비루스로부터 보호하는 갱신된 항비루스프로그람을 정기적으로 받는다.

수자면역체계의 성공여부는 새로운 비루스변종을 검출하는 비루스해석기의 능력에 관계된다. 새로운 비루스들을 항상 해석하고 조종함으로써 위협을 받지 않게 수자면역쏘프트웨어를 자주 갱신하여야 한다.

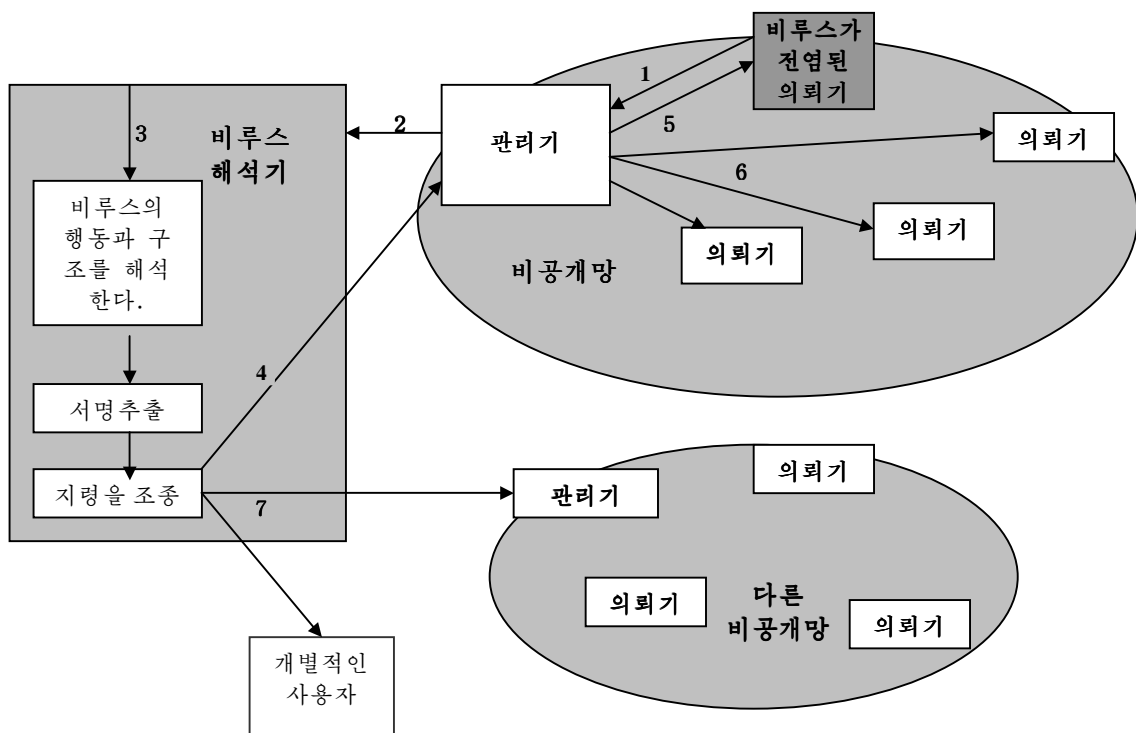



그림 15-11. 수자면역체계

참고문헌

침입자에 대한 더 구체적인 설명은 참고문헌 [STOL89]에 있다. 참고문헌 [STER92]에서는 가치 있는 침입력사를 제공하며 침입자의 기술과 목표를 론한다. 참고문헌 [DENN90]에는 침입자를 론하는 많은 기본론문들의 재판문헌들이 있다.

참고문헌 [HOFF90]과 [DENN90]에는 비루스와 웜을 론하는 기본론문들이 많다. 참고문헌 [COHE94])에는 비루스와 항비루스기술에 대한 기술적설명이 있다. 참고문헌 [FORR97], [KEPH97],[NACH97]에는 비루스에 대한 문제의 개요를 주고 있다.

COHE94 Cohen,F. A *Short Course on Computer Viruses*. New York:Wiley,1994

DENN90 Dening,P. ,editor. *Computer Under Attack:Intruders,Worms,and Viruses*. Reading,MA:Addison-Weasely,1990.

FORR97 Forrect,S. ;Hofmeyr,S. :and Somayaji,A. 《Computer Immunology.》 *Communications of the ACM*,Octerber 1997.

HOFF90 Hoffman,L. ,editor. Rogue *Programs*:*Viruses*,*Worms*,*snd Trojan Horses*. New York:Van Nostrand Reinhold,1990

KEPH97 Kephart, J. ;Sorkin, G. ;Chess, D. ;and White, S. 《Fighting Computer Viruses. 》*Scientific American*, November 1997
NACH97 Nachenberg, C. 《 Computer Virus-Antivirus coevolution. 》 *Communications of the ACM*, January 1997
STER92 Sterling, B. The *Hacker Crackdown:Law and Disorder on the Electronic Frontier*. New York:Bantam, 1992.
STOL89 Stoll, C. *The Cuckoo's Egg*. New York:Doubleday, 1989.

참고할 Web싸이트들

- **CERT조종쎈터:** 대상과제표준탐색방어회사(Defense Advanced Research Projects Agency)가 운영하는 콤퓨터긴급대책팀에서 나온 기구이다. 싸이트는 인터네트보안위협, 약점, 공격통계량에 대한 좋은 정보를 제공한다.
- **항비루스직결:** 비루스정보에 대한 IBM의 싸이트로서 매우 편리하다.

문 제

1. 26개의 자모기호들중 4개의 기호를 취한 조합에서 통과암호를 선택한다고 하자. 그리고 적수는 초당 통과암호를 한개씩 시험할수 있다고 가정하자.
   ㄱ) 매 시험이 실현될 때까지 적에게로의 반결합은 없다고 가정한다. 정확한 통과암호를 발견하는데 얼마만한 시간이 걸리겠는가?
   ㄴ) 매번 부정확한 문자가 입력될 때 적에게 오유를 통보하는 반결합체계라고 하면 이때 정확한 통과암호를 발견하는데 얼마만한 시간이 걸리겠는가?
2. 어떤 방법으로 길이 k인 원천원소들을 길이 p인 목적원소들로 넘긴다고 가정하자. 매 수자가 r개의 값들중 한개를 취할수 있다고 하면 원천원소의 수는 r^k이며 목적원소의 수는 보다 작은 수 r^p이다. 개개의 원천원소 x_i는 개개의 목적원소 y_j에로 넘어 간다.
   ㄱ) 적수가 단번에 정확한 원천원소를 선택할수 있는 확률은 얼마인가?
   ㄴ) 같은 목적원소 y_i로 되는 다른 원천원소 x_k ($x_i \neq x_k$)를 적수가 만들어 낼수 있는 확률은 얼마인가?
   ㄷ) 적수가 단번에 정확한 목적원소를 만들수 있는 확률은 얼마인가?
3. 음성통과암호생성기는 문자가 6개인 통과암호에 대하여 우연적으로 두개의 토막을 선택한다. 매 토막의 형태는 CVC(자음, 모음, 자음)이며 여기서 V=<a, e, i, o, u> 이고 $C=\overline{V}$ 이다.
   ㄱ) 통과암호는 모두 몇개인가?
   ㄴ) 적수가 통과암호를 정확히 추측할 확률은 얼마인가?
4. 통과암호로서는 95개의 볼수 있는 아스키부호만을 리용하며 모든 통과암호들의 길이는 10문자이라고 가정한다. 통과암호크랙커는 초당 640만번의 속도로 암호화한

다고 가정한다. UNIX체계에서 모든 가능한 통과암호들을 철저히 남김없이 검사하는데 얼마나 긴 시간이 걸리겠는가?

5. UNIX의 통과암호체계가 위험하므로 SunOS-4.0문서화에서는 통과암호파일을 제거하고 공개적으로 읽을수 있는 /etc/publickey라고 부르는 파일로 교체할것을 권고하고 있다. 사용자 A의 파일에로의 입력자료는 사용자의 식별자 ID_A,사용자의 공개열쇠 KU_a,대응하는 비밀열쇠 KR_a이다. 이 비밀열쇠는 사용자의 가입통과암호 P_a로부터 만든 열쇠로 DES를 리용하여 암호화한다. A는 체계에 가입할 때 KR_a를 얻기 위해 $E_{P_a}[KR_a]$를 복호한다.

   ㄱ) 이때 체계는 P_a가 정확히 입력되였다는것을 검증한다. 어떻게 검증할수 있는가?

   ㄴ) 적수는 이 체계를 어떻게 공격할수 있는가?

6. UNIX의 통과암호에서 리용하는 암호도식은 한 방향성이다. 이것은 거꾸로 할수 없다. 따라서 이것은 통과암호의 암호화라기보다도 사실상 하쉬부호라고 말하는것이 정확하지 않겠는가?

7. UNIX의 통과암호도식에서는 쏠트로서 4096의 인수를 리용함으로써 추측의 어려움성을 증가시킨다고 할수 있다. 그러나 쏠트는 해당 암호문통과암호로서 같은 입력자료에 평문으로 기억된다. 따라서 이 두개의 기호들은 공격자에게 알려 질수 있으나 추측되지 말아야 한다. 왜 쏠트가 보안을 증가시킨다고 주장하는가?

8. 당신이 앞의 문제를 성과적으로 대답하고 쏠트의 의미를 리해하였다고 가정하면 여기에 다른 물음이 있다. 말하자면 쏠트의 크기를 24 혹은 48bit로 극단적으로 증가시킴으로써 모든 통과암호크랙커들을 완전히 막는것이 가능한가?

9. 15.1에서 론의한 블룸려과기를 생각하자. K:하쉬함수의 수, N:하쉬표에서 비트수, D:사전의 단어수라고 정의한다.

   ㄱ) 령으로 초기화된 하쉬표에서 확장된 비트들의 수는

$$\phi = (1 - \frac{k}{N})^D$$

   로서 표현된다는것을 증명하시오.

   ㄴ) 사전에 없는 입력자료단어를 사전에 있는것으로 틀리게 받아 들일 확률은

$$P = (1-\phi)^k$$

   이라는것을 증명하시오.

   ㄷ) 앞의 식은

$$P \approx (1-e^{-kD/N})^k$$

   로서 근사화할수 있다는것을 증명하시오.

10. 그림 15-8의 비루스프로그람에는 약점이 있다. 그것은 무엇인가?

제16장. 방화벽

방화벽은 광지역망과 인터네트를 통한 외부세계에로의 접근을 제공하는 동시에 망에 기초한 보안위협으로부터 국부체계나 망체계들을 보호하는 효과적인 수단으로 될수 있다.

방화벽의 기능과 설계원리를 개괄하는것으로부터 이 장을 시작한다. 다음으로 방화벽자체의 보안문제 특히 신용체계의 개념과 안전한 조작체계에 대하여 론의한다.

16.1 방화벽의 설계원리

회사나 정부기관 기타 기관들에서의 정보체계는 부단히 발전하여 왔다.

- 직결된 많은 말단들을 지원하는 중앙주프레임을 가지는 중앙자료처리체계
- PC와 말단들을 서로 접속하고 이것들을 주프레임에 접속하는 국부망(LAN)
- PC들, 봉사기들 그리고 대체로 한개 혹은 두개의 주프레임을 서로 접속하는 많은 LAN들로 구성되는 전제망
- 비공개광지역망(WAN)에 의해 서로 접속되여 있는 다중의 지리학적으로 분산되여 있는 전제망들로 이루어 진 대규모망
- 각이한 전제망들이 인터네트에 모두 련결되면서도 비공개WAN에 의해 접속될수 있거나 접속될수 없는 인터네트접속성

대부분의 기관들은 더는 인터네트접속을 마음대로 할수 없게 되였다. 가치 있는 정보들과 봉사들은 기관에서 필수적인것으로 된다. 더우기 개별적인 사용자들은 인터네트접근을 요구하거나 바라며 이것이 자기들의 LAN을 통해 제공되지 않으면 자기들의 PC로부터 인터네트봉사제공자(ISP)에게 전화를 걸수 있는 자격을 리용할것이다. 그러나 인터네트접근은 기관에 리익을 제공해 주는 한편 외부세계가 국부망에 련결되여 호상작용할수 있게 한다. 이것은 기관에 위협을 준다. 전제망에 있는 개개의 봉사기에 침입보호와 같은 강력한 보안특성을 부여하는것은 실천적인 방법으로는 되지 못한다. UNIX 그외에 Windows 95, 98, NT의 각이한 판본을 실행하는 수백개 지어 수천개의 체계들을 가지고 있는 망을 생각한다. 보안결함을 발견하였을 때 영향을 받을 가능성이 있는 개개의 체계는 이 결함을 고치기 위해 개량하여야 한다. 점차적으로 많이 취하는 대안이 방화벽이다. 조종련결을 확립한 다음 외부보안벽이나 돌출부를 설치하기 위해 방화벽을 전제망과 인터네트사이에 배치한다. 돌출부의 목적은 전제망을 인터네트에 기초한 공격으로부터 보호하는것이며 보안과 검열을 맡은 단일조절점을 제공하는것이다. 방화벽은 단일콤퓨터체계로 되거나 방화벽의 기능을 수행하기 위해 서로 협동하는 두개 혹은 그이상의 체계모임으로 될수 있다.

이 절에서는 방화벽의 형태들에 대한 고찰로부터 시작하여 가장 일반적인 방화벽의 몇가지 구성방법을 고찰한다.

방화벽의 특성

참고문헌 [BELL94]에는 다음과 같은 방화벽의 설계목표가 제시되였다.

1. 내부로부터 외부에로 혹은 그 반대방향으로의 모든 통신은 방화벽을 통과하여야 한다. 이것은 방화벽을 거치지 않는 국부망에 대한 모든 접근을 물리적으로 차단함으로써 달성할수 있다. 여러가지 구성방법들이 있을수 있다.
2. 국부보안방책에 의해서 정의되는 권한을 가진 통신만이 통과할수 있게 된다. 여러가지 형태의 방화벽들의 리용에 기초하여 여러가지 형태의 보안방략들이 실현된다.
3. 방화벽은 그 자체가 침투에 대한 면역을 가진다. 이것은 안전한 조작체계를 가진 신용체계를 리용한다는것을 의미한다. 이 문제는 2절에서 론의한다.

참고문헌[SMIT97]에서는 접근조종과 싸이트의 보안방략을 강화하는데 방화벽들을 리용하는 일반적인 4가지 기술을 서술하였다. 원래 방화벽들은 기본적으로 봉사조종에 초점을 두었지만 그후 그것들은 모두 4가지 기술을 제공하는데로 발전하였다.

- **봉사조종:**이것은 접근할수 있는 인터네트의 봉사형태를 결정한다. 방화벽은 IP주소나 TCP포구번호에 기초하여 통신을 려과하거나 방화벽을 통과하기전에 매 봉사요구를 받아 해석하는 대리쏘프트웨어를 제공할수 있으며 Web나 전자우편봉사와 같은 봉사기쏘프트웨어에 직접 가입할수 있다.
- **방향조종:**이것은 개별적인 봉사요구들이 방화벽을 통해 접수되여 흐르게 되는 방향을 결정한다.
- **사용자조종:**이것은 봉사에로의 접근을 조종하는데 그에 따라 사용자는 접근하려한다. 이 기능은 일반적으로 방화벽의 돌출부안에 있는 사용자(국부사용자)에게 적용된다. 그것은 또한 외부사용자로부터 들어 오는 통신에 적용될수도 있으며 후자는 IPSec(13장)에서 제공한것과 같은 안전한 인증기술의 어떤 형식을 요구한다.
- **거동조종:**이것은 개별적인 봉사들의 리용되는 방법을 조종한다. 례를 들어 방화벽은 전자우편을 려과하여 스팸을 제거할수 있으며 국부Web봉사에서의 일부 정보에 만 외부접근을 허용한다.

방화벽의 형태와 구성을 구체적으로 보기전에 방화벽에서 기대할수 있는것이 무엇인가를 개괄해 보자. 다음의 기준들은 방화벽의 범위내에 속한다.

1. 방화벽은 권한이 없는 사용자들을 보호된 망밖에 있게 하며 공격당하기 쉬운 봉사들이 망에 들어 오거나 나가는것을 금지하는 단일조절점을 정의하며 IP기만이나 경로조종공격과 같은 각이한 공격으로부터 보호한다. 단일조절점의 리용은 단일체계나 체계묶음에서 보안능력들이 통합정리되므로 보안관리를 단순화한다.
2. 방화벽은 보안과 관련한 사건들을 감시조종하기 위한 위치를 제공한다. 방화벽체계에서는 검열과 경보가 실현될수 있다.
3. 방화벽은 관련되는 보안이 없는 여러 인터네트기능들에 편리한 가동환경으로 된다. 이러한 기능들로는 국부주소를 인터네트주소로 넘기는 망주소번역기와 인터네트사용을 검열하거나 기록하는 망관리기능을 들수 있다.

4. 방화벽은 인터네트규약보안(IPSec)에 대한 가동환경으로서 봉사할수 있다. 13장에서 서술한 통로방식자격을 리용하면 방화벽은 가상적인 비공개망을 실현하는데 리용할수 있다.

방화벽에는 다음과 같은 제한성들이 있다.

1. 방화벽은 그것을 우회하는 공격들을 막을수 없다. 내부체계는 ISP에 대한 접속능력을 전화접속해제할수 있다. 내부LAN은 려행자들과 원격콤퓨터들의 전화접속능력을 제공하는 모뎀묶음을 지원할수 있다.
2. 방화벽은 불만을 품은 사용자나 자기도 모르게 외부공격자에게 역리용 당하는 사용자와 같은 내부위협에 대하여 보호할수 없다.
3. 방화벽은 비루스에 전염된 프로그람이나 파일이 전송되는것을 막을수 없다. 돌출부안에서 지원되는 응용프로그람들과 조작체계의 다양성으로 하여 방화벽이 비루스에 전염되여 들어 오는 모든 파일이나 전자우편, 통보문을 조사하는것은 비현실적이며 거의나 불가능하다.

방화벽의 형태

참고문헌 [SEME96]에 기초하여 그림 16-1에 방화벽의 세가지 일반형태 즉 파케트려파기, 응용준위관문, 회선준위관문에 대하여 보여 주었는데 그 형태들을 차례로 고찰하면 다음과 같다.

파케트려과경로조종기

파케트려과경로조종기는 들어 오는 매 IP파케트에 대하여 규칙모임을 적용한후 파케트를 전송하거나 거부한다. 경로조종은 일반적으로 두 방향(내부망으로부터와 내부망에로)에서 오는 파케트들을 려파한다. 려파규칙들은 원천지와 목적지의 IP주소,IP규약마당(전송규약을 정의하는)과 TCP 혹은 UDP포구번호(SNMP 혹은 TELNET와 같은 응용프로그람을 정의하는)를 포함하는 IP마당들과 전송머리부(즉 TCP나 UDP)에 기초하고 있다.

파케트려과는 일반적으로 IP 혹은 TCP머리부에 있는 마당들을 대조하는데 기초한 규칙목록으로 설정한다. 어느한 규칙과 대조될 때 그 규칙은 파케트를 전송하거나 거부하겠는가를 결정하게 된다. 그 어느 규칙과도 대조되지 않으면 기정동작을 취한다. 두가지 기정방책이 있을수 있다.

- Default = discard: 명백히 허락되지 않은것은 금지한다.
- Default = forword: 명백히 금지되지 않은것은 허락한다.

기정거부방책이 더 신중하다. 초기에는 무엇이나 다 블로크화되며 봉사들은 경우에 따라 토대에 첨가되여야 한다. 이 방책은 방화벽을 방패처럼 여기는 사용자들에게는 매우 명백하다. 기정전송방책은 말단사용자의 편리성을 도모하지만 약한 보안을 제공하며 보안관리자는 요컨대 알려 질 때마다 매번 새로운 보안위협에 대응하여야 한다.

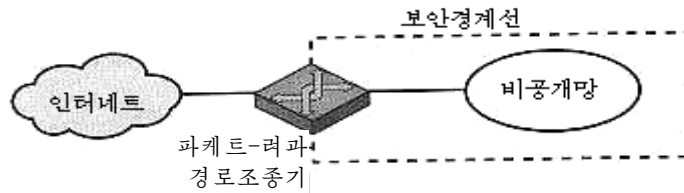
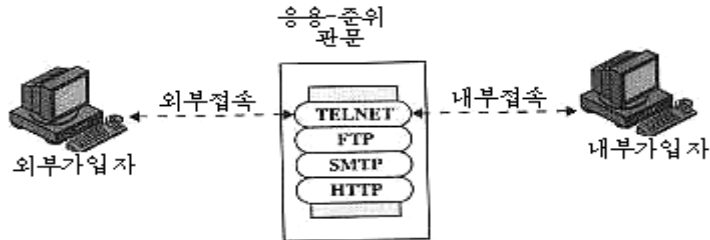
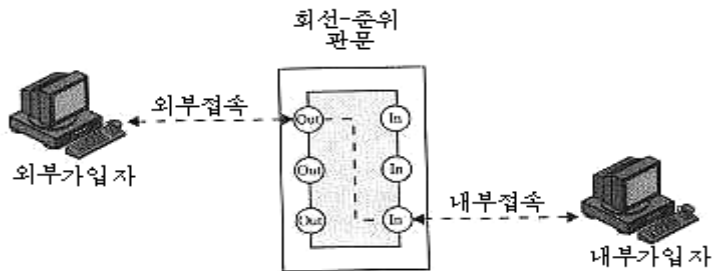
참고문헌 [BELL94]에 주어 진 표 16-1이 파케트려과규칙모임의 한가지 실례이다. 매 모임에서 규칙들은 우에서 아래로 적용된다. 마당에서 《*》은 아무것이나 다 대조되는 윌드카드지정자이다. 기정=거부방책이 성립한다고 가정한다.

ㄱ) 파케트-려과경로조종기

ㄴ) 응용-준위관문

ㄷ) 회선-준위관문

그림16-1. 방화벽의 형태

ㄱ) 돌아 가는 우편은 관문가입자에 대해서만 허락된다(포구 25는 SMTP가입을 위한 것이다). 그러나 개개의 외부가입자 즉 SPIGOT로부터 오는 우편은 그 가입자가 전자우편통보문들에 용량이 큰 파일들을 보내는 리력을 가지고 있기때문에 블로크화된다.

ㄴ) 이것은 기정방책에 대한 명시적서술문이다. 모든 규칙모임들은 마지막규칙으로서 이 규칙을 무조건 포함한다.

ㄷ) 이 규칙모임은 임의의 내부가입자가 외부가입자에게 우편을 보낸다는것을 설명한다. 목적지포구가 25인 TCP파케트는 목적지에 있는 SMTP봉사기에로 발송된다. 이 규칙에서 문제는 SMTP수신때 포구 25를 리용하는것이 유일한 결점이라는것이다. 이 규칙에 씌여 진것처럼 공격자는 TCP원천포구번호 25를 가지는 파케트를 보냄으로써 내부기계에 접근할수 있다.

ㄹ) 이 규칙모임은 ㄷ에서 달성되지 못한 목적결과를 달성하게 한다. 규칙들은 TCP접속기능의 우점을 리용한다. 일단 접속이 설정되면 TCP토막의 ACK기발은 다른 곳에서 보낸 응답토막에 설정된다. 그러므로 이 규칙모임은 원천지IP주소가 지정

된 내부가입자목록중 하나이고 목적지의 TCP포구번호가 25인 IP파케트들을 허락한다는것을 나타낸다. 또한 원천지포구번호가 25이고 TCP토막에 ACK기발을 포함하는 가입파케트를 허락한다. 이 규칙들을 명백히 정의하기 위해서는 원천지와 목적지체계를 명백히 지정하여야 한다는것에 주의하시오.

ㅁ) 이 규칙모임은 FTP(파일전송규약)접속을 조종하는 한가지 방식이다. FTP로는 두가지 TCP접속이 리용된다. 즉 파일전송을 설치하는 조종접속과 실제적인 파일전송을 위한 자료접속이다. 자료접속에서는 전송을 위해 동적으로 할당되는 다른 포구번호가 리용된다. 대부분의 봉사기들, 그로부터 대부분의 공격목표들은 작은 번호를 가지는 포구에 있으며 대부분의 출발호출은 보통 1023이상의 높은 번호를 가지는 포구를 리용하게 된다. 즉 규칙모임은 다음의것을 허락한다.

- o 내부적으로 발생하는 파케트들
- o 내부기계에 의해 새로 가입한 접속에 대한 응답파케트들
- o 내부기계에서 높은 번호를 가지는 포구에 대해 지정되는 파케트들

표 16-1. 파케트-려과 실례

동작	우리의 가입자	포구	그들의 가입자	포구	설명
블로크	*	*	SPIGOT	*	이 사람들은 믿을수 없다.
허락	OUR-GW	25	*	*	우리의 SMTP포구에 접속

ㄱ)

동작	우리의 가입자	포구	그들의 가입자	포구	설명
블로크	*	*	*	*	기정

ㄴ)

동작	우리의 가입자	포구	그들의 가입자	포구	설명
허락	*	*	*	25	그들의 SMTP포구에 접속

ㄷ)

동작	원천지	포구	목적지	포구	기발들	설명
허락	{우리의 가입자}	*	*	25		그들의 SMTP포구에 대한 우리의 파케트
허락	*	25	*	*	ACK	그들의 응답

ㄹ)

동작	원천지	포구	목적지	포구	기발들	설명
허락	{우리의 가입자}	*	*	*		우리의 송신호출
허락	*	*	*	*	ACK	우리의 호출에 대한 응답
허락	*	*	*	>1024		비봉사기에로의 통신량

ㅁ)

이 도식은 체계를 적당한 포구번호만 리용하여 구성할것을 요구한다.

규칙모임 ㅁ은 파케트려파준위에서 응용프로그람을 처리하기가 힘들다는것을 지적한다. FTP와 그런 류형의 응용프로그람들을 취급하는 다른 방법은 응용프로그람준위의 관문이다.

파케트려파경로조종기의 한가지 우점은 그것이 간단하다는것이다. 또한 파케트려파는 일반적으로 사용자들이 알기 쉬우며 매우 빠르다. 결함은 파케트려파규칙을 정확히 설정하는것이 힘들며 인증이 필요하다는것이다.

참고문헌 [SEME96]에서는 파케트-려파경로조종기에 대해서 있을수 있는 일부 공격과 이에 적합한 대응수단들을 서술하였다. 즉

- **IP주소속이기:** 침입자는 외부에서 내부가입자의 주소가 들어 있는 원천지IP주소마당으로 파케트를 전송한다. 공격자는 주소를 속여 넘기는 방법으로 명백히 신용되는 내부가입자들의 파케트들을 허락하는 단순한 원천지주소보안을 취하는 체계에 침입하려고 한다. 대응수단은 파케트가 외부대면을 통해 도착하였다면 내부원천지주소로 그 파케트들을 거부하는것이다.
- **원천경로조종공격:** 원천국은 원천지경로조종정보를 분석하지 않는 보안수단을 우회하기를 바라면서 파케트가 인터네트로 넘어 갈 때 통과해야 할 경로를 서술한다. 대응수단은 이 선택을 리용하는 모든 파케트들을 거부한다.
- **작은 토막공격:** 침입자는 매우 작은 토막을 만들고 분리된 파케트토막에서 TCP머리부정보를 절취하기 위해 IP토막화선택을 리용한다. 이 공격은 TCP머리부정보에 관한 려과규칙들을 우회하도록 설계된다. 공격자는 첫번째 토막만이 려과경로조종에 의해 조사되고 나머지토막은 그냥 통과될것을 바란다. 작은 토막공격은 규약형태가 TCP이고 IP토막편차가 1과 같은 모든 파케트들을 거부함으로써 막을수 있다.

응용준위관문

대리봉사기라고도 부르는 응용준위관문은 응용준위전송의 **대리봉사기**로서 작용한다(그림16-1의 ㄴ). 사용자는 Telnet나 FTP와 같은 TCP/IP응용을 리용하는 관문에 접속하며 관문은 접근할수 있는 원격가입자의 이름을 사용자에게 요구한다. 사용자가 응답하고 정당한 사용자ID와 인증정보를 제공할 때 관문은 원격가입자의 응용을 접속하며 두 말단지점사이의 응용자료를 포함하는 TCP토막들을 중계한다. 관문이 명확한 응용프로그람에 대한 대리코드를 실현하지 않으면 봉사는 지원되지 않으며 방화벽을 통해 전송할수 없다. 더우기 관문은 망관리자가 다른 모든 특징을 거부하지만 접수할수 있다고 생각하는 응용프로그람의 명백한 특징만을 지원하도록 구성할수 있다.

응용준위관문은 파케트려파보다 더 안전할수 있다. TCP와 IP준위에서 허락되거나 금지되는 여러개의 가능한 조합의 취급이 아니라 응용준위관문은 몇개의 허락할수 있는 응용들만을 세밀히 조사할것을 요구한다. 게다가 응용준위에서는 수신한 모든 통신량을 검열하고 기록(log)하는것이 쉽다.

이러한 관문형태의 주요한 결함은 접속할 때마다 매번 추가적인 부가처리가 있다는것이다. 사실상 련결점에 관문을 가지고 있는 말단사용자들사이에는 련결된 두개의 접속이 있으며 관문은 두 방향에서 모든 통신을 보내고 조사하여야 한다.

회선준위관문

방화벽의 세번째 형태는 회선준위관문이다(그림 16-1의 ㄷ). 이것은 단독체계로 될수 있거나 혹은 어떤 응용에 대해서는 응용준위관문이 수행하는 특수한 기능으로 될수 있다. 회선준위관문은 말단과 말단사이의 TCP접속을 허락하지 않는다. 그대신 관문은 두개의 TCP접속 즉 자기자신과 내부가입자우의 TCP사용자와의 접속 및 자기자신과 외부가입자에 있는 TCP사용자와의 접속을 설정한다. 일단 두 접속이 확립되면 관문은 일반적으로 내용을 조사하지 않고 한개의 접속으로부터 다른 접속에로 TCP토막을 중계한다. 보안기능은 어느 접속들을 허락하겠는가를 결정하는것이다.

회선준위관문의 일반적인 리용은 체계관리자가 내부사용자들을 믿는 경우이다. 관문은 안으로 들어 오는 접속에 대한 응용준위 혹은 대리봉사와 밖으로 나가는 접속에 대한 회선준위기능을 지원하도록 구성할수 있다. 이 구성에서 관문은 금지된 응용자료가 들어오는 경우 이것을 조사하는 처리를 진행할수 있지만 나가는 자료에 대해서는 이러한 처리를 진행하지 않는다.

회선준위관문의 실현실례는 **SOCKS**제품이다(참고문헌 [KOBL92]). SOCKS의 제5판은 RFC 1928에 정의되여 있다. RFC는 다음과 같은 방식으로 SOCKS를 정의한다.

SOCKS에 서술된 통식규약은 망방화벽의 봉사들을 편리하게 그리고 안전하게 리용하기 위해 TCP와 UDP령역에서 의뢰기-봉사기응용프로그람에 프레임워크를 제공하도록 설계한다. 규약은 개념적으로는 응용층과 전송층사이의 "쐐기층"이며 이것만으로는 ICMP통보문을 전송하는것과 같은 망층관문봉사들을 하지 못한다.

SOCKS는 다음의 구성요소를 포함한다.

- UNIX에 기초한 방화벽에서 실행하는 SOCKS봉사기
- 방화벽에 의해 보호되는 내부가입자에서 실행하는 SOCKS의뢰기서고
- FTP와 TELNET와 같은 여러가지 표준의뢰기프로그람의 SOCKS판본들. SOCKS규약은 일반적으로 SOCKS서고에 있는 대응하는 밀봉루틴을 리용하여 TCP에 기초한 의뢰기응용프로그람을 재콤파일하거나 재련결함으로써 실현된다.

TCP에 기초한 의뢰기가 방화벽을 통해서만 도달할수 있는 객체에 접속하려고 할 때 그것은 SOCKS봉사기체계에서 대응하는 SOCKS포구에 TCP접속을 하여야 한다. SOCKS봉사는 TCP포구1080에 위치하고 있다. 접속요구가 성공되면 의뢰기는 리용할 인증방법에 대한 교섭을 시작하며 선택한 방법으로써 인증한 다음 중계요구를 보낸다. SOCKS봉사기는 요구를 평가하고 적당한 접속을 확립하든가 확립하지 않는다. UDP교환도 류사한 방식으로 조종한다. 본질적으로 TCP접속은 UDP토막을 송신하고 수신하는 사용자를 인증하기 위해 열려 져 있으며 UDP토막은 TCP접속이 열려 져 있는동안 전송된다.

요새가입자

요새가입자는 망보안의 림계강도점으로서 방화벽관리자에 의해 식별되는 체계이다. 일반적으로 요새가입자는 응용준위나 회선준위관문에 대한 가동환경으로 봉사한다. 요새가입자의 일반적인 특성은 다음과 같다.

- 요새가입자의 하드웨어가동환경은 믿음직한 체계인 안전한 조작체계판본을 실행한다.
- 망관리자가 본질적이라고 생각하는 봉사들만이 요새가입자에 설치된다. 이것들은 Telnet, DNS, FTP, SMTP, 사용자인증과 같은 대리응용프로그람을 포함한다.
- 요새가입자는 사용자가 대리봉사에 접근하기전에 추가적인 인증을 요구할수 있다. 게다가 매 대리봉사는 사용자접근을 허가하기전에 그것들자신의 인증을 요구한다.
- 매 대리인은 표준적인 응용지령모임의 부분모임만을 지원하도록 구성된다.
- 매 대리인은 명백한 가입자체계에만 접근하도록 구성된다. 이것은 제한된 지령/특징모임이 보호된 망에 있는 체계의 부분모임에만 적용된다는것을 의미한다.
- 매 대리인모듈은 망보안을 위해 특별히 설계된 매우 작은 쏘프트웨어제품이다. 이것은 비교적 간단하므로 이와 같은 모듈에 대한 보안결함을 쉽게 검사할수 있다. 실례로 전형적인 UNIX의 전자우편응용프로그람은 20000개이상의 코드행을 포함하지만 전자우편대리프로그람은 1000개이하의 코드행을 포함한다(참고문헌[SEME96]).
- 매 대리인은 요새가입자의 다른 대리인과 독립이다. 임의의 대리인의 조작상문제가 있거나 앞으로 약점이 발견되여 그것이 다시 설치될 때 다른 대리응용프로그람의 조작에 영향을 미치지 않게 할수 있다. 또한 사용자집단이 새로운 봉사에 대한 지원을 요구하면 망관리자는 요새가입자에 요구하는 대리인을 쉽게 설치할수 있다.
- 대리인은 일반적으로 초기구성파일을 읽는 경우외에는 디스크에 접근할수 없다. 그러므로 침입자가 요새가입자에 트로이목마탐지기 혹은 다른 위험한 파일들을 설치할수 없게 된다.
- 매 대리인은 비특권사용자로서 요새가입자의 비공개적이며 안전한 등록부에서 실행된다.

방화벽의 구성

단일파케트려과경로조종기 혹은 단일관문과 같은 단일체계로서 간단히 구성되는외에(그림 16-1) 더 복잡한 구성들도 가능하다. 참고문헌[STEM96]의 그림에 기초한 그림 16-2는 3가지 일반적인 방화벽의 구성을 보여 준다. 그것을 차례로 고찰하자.

차단가입자방화벽, 단일홈요새의 구성(그림 16-2의 ㄱ)에서 방화벽은 두 체계 즉 파케트려과경로조종기와 요새가입자로 이루어 진다. 일반적으로 경로조종기는 다음과 같이 구성된다.

1. 인터네트로부터의 통신에 대해서는 요새가입자에게로 지정된 IP파케트들만 받아들인다.
2. 내부망으로부터의 통신에 대해서는 요새가입자로부터 오는 IP파케트들만 내보낸다.

요새가입자는 인증과 대리인기능을 수행한다. 이 구성은 단순한 파케트려과경로조종기나 응용준위관문보다 두가지 리유로 보다 좋은 보안을 할수 있다. 첫째로, 이 구성이

려과의 파케트준위와 응용준위를 다 실현함으로써 보안방책의 정의에서 유연성을 현저히 증가시킨다는것이다. 둘째로, 침입자는 내부망보안을 약화시키기 위해 일반적으로 두개의 분리된 체계에 침투하여야 한다는것이다.

이 구성도 직접적인 인터네트접근을 제공할 때 유연성을 가진다. 례를 들어 내부망은 고준위보안이 요구되지 않는 Web봉사기와 같은 공개정보봉사기를 포함할수 있다. 이 경우에 경로조종기는 정보봉사기와 인터네트사이에 직접 통신하도록 구성할수 있다.

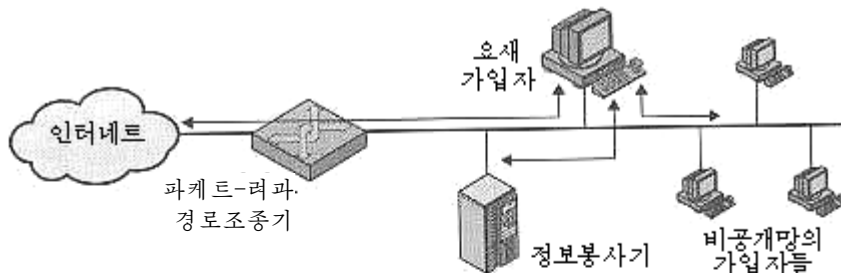
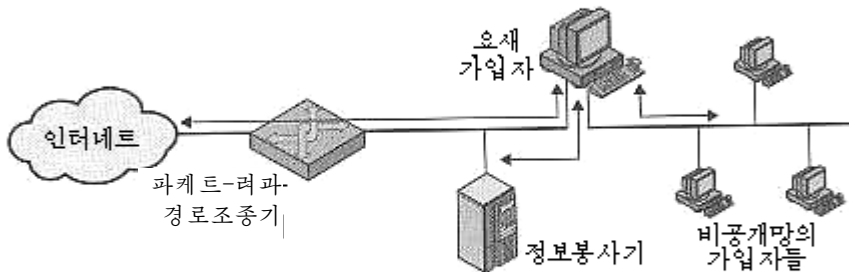
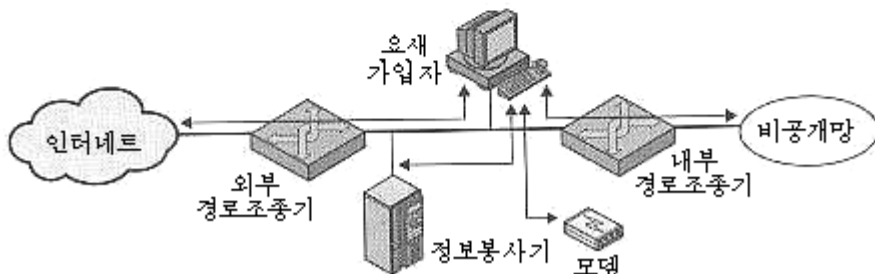
방금 서술한 단일홈구성에서 파케트려과조종기가 매우 위태롭게 되면 통신은 비공개망에서 인터네트와 다른 가입자사이에 경로조종기를 통해 직접 진행할수 있을것이다. **차단된 가입자방화벽,쌍홈요새**구성은 보안위반과 같은 침입을 물리적으로 못하게 한다(그림 16-2의 ㄴ). 앞의 구성에서의 쌍보안층의 우점을 여기서도 찾아 볼수 있다. 또한 정보봉사기나 다른 가입자들은 이것이 보안방책과 부합되면 경로조종기와 직접 통신하게 할수 있다.

ㄱ)차단된 가입자방화벽체계(단일홈요새가입자)

ㄴ)차단된 가입자방화벽체계(쌍홈요새가입자)

ㄷ)차단된 부분망 방화벽체계

그림 16-2. 방화벽의 구성

그림 16-2의 ㄷ의 **차단된 부분망방화벽**의 구성은 지금까지 고찰한것들가운데서 가장 안전하다. 이 구성에서는 두개의 파케트려파경로조종기 즉 요새가입자와 인터네트사이에 있는것과 요새가입자와 내부망사이에 있는것을 리용하였다. 이 구성은 고립된 부분망을 만드는데 이 부분망은 단순히 요새가입자로 이루어 질수 있지만 한개 혹은 그이상의 정보봉사기들과 전화접속자격을 주는 모뎀들을 포함할수 있다. 일반적으로 인터네트와 내부망은 둘다 차단된 부분망에서 가입자에 대한 접근을 가지지만 차단된 부분망을 통한 통신은 블로크화된다. 이 구성은 다음과 같은 우점을 가진다.

- 침입자를 막는 3개의 방어준위가 있다.
- 외부경로조종기는 인터네트에 차단된 부분망의 존재에 대해서만 알려 준다. 따라서 내부망은 인터네트에 공개되지 않는다.
- 마찬가지로 내부경로조종기는 내부망에 차단된 부분망의 존재에 대해서만 알려 준다. 따라서 내부망체계는 인터네트에 대한 직접경로를 구성할수 없다.

16.2 신용체계

침입자와 불순한 프로그람을 막는 체계의 능력을 높이는 한가지 방법은 신용체계기술을 실현하는것이다. 이 절에서는그에 대한 간단한 개요를 준다. 먼저 자료접근조종의 일부 기본개념을 고찰하는것으로부터 시작한다.

자료접근조종

가입하는데 성공한 사용자는 하나의 가입자나 응용프로그람 혹은 이것들의 모임에 접근할수 있다.일반적으로 자기 자료기지에 민감한 자료를 포함하는 체계에 대해서는 이것만으로는 불충분하다. 사용자접근조종절차를 통하여 사용자는 체계와 결합될수 있다. 매 사용자와 관련한 허용된 조작과 파일접근을 기록한 개요파일이 체계에 있을수 있다. 조작체계는 다음 사용자개요에 기초한 규칙들을 실시할수 있다. 그러나 자료기지관리체계는 특정의 레코드나 지어 레코드들에 대한 접근도 조종하여야 한다. 례를 들어 행정관리에서는 아무사람이나 회사성원의 목록을 얻을수 있지만 선택된 사람들만이 월생활비정보에 접근할수 있다. 이것은 한 계단 더 상세한 준위이다. 조작체계는 사용자에게 그이상 더는 보안검사가 없는 파일에로의 접근이나 응용프로그람의 리용을 허락하게 하는 반면에 자료기지관리체계는 매 개별적인 사람의 접근시도에 대해 결정하여야 한다. 그 결정은 사용자의 신분뿐아니라 접속하는 자료의 특정한 부분 지어 이미 사용자에게 폭로된 정보에도 관계될것이다.

파일이나 자료기지관리체계가 리용하는 접근조종의 일반적인 모형은 **접근행렬**이다(그림 16-3의 ㄱ). 이 모형의 기본요소는 다음과 같다.

- **주동체: 객체**에 접근할수 있는 실체이다. 일반적으로 주동체의 개념은 처리와 같다. 임의의 사용자나 응용프로그람은 실제로 이것들을 나타내는 처리에 의해 객체에 대한 접근을 얻는다.
- **객체:** 접근이 조종되는 임의의것. 실례로 파일, 파일의 부분, 프로그람, 기억기토막을 들수 있다.
- **접근권:** 주동체에 의해 대상에 접근하는 방법. 실례로 읽기, 쓰기, 실행을 들수 있다.

	프로그람 1	…	토막 A	토막 B
처리공정 1	읽기 실행		읽기 쓰기	
처리공정 2				읽기
. . .				

ㄱ) 접근행렬

프로그람에 대한 접근조종목록: 처리공정 1(읽기, 실행)
토막 A에 대한 접근조종목록: 처리공정 1(읽기, 쓰기)
토막 B에 대한 접근조종목록: 처리공정 2(읽기)

ㄴ) 접근조종목록

처리공정 1에 대한 자격목록: 프로그람 1(읽기, 실행) 토막 A(읽기, 쓰기)
처리공정 2에 대한 자격목록: 토막 B(읽기)

ㄷ) 자격목록

그림 16-3. 접근조종구조

행렬의 한 축은 자료접근을 시도할수 있는 식별된 주동체들로 구성되여 있다. 일반적으로 이 목록은 접근이 사용자대신에 혹은 사용자와 함께 말단, 가입자, 응용프로그람에 대해 조종될수 있어도 개별적인 사용자나 사용자그룹으로서 구성된다. 다른 축은 접근할수 있는 대상들을 렬거한다. 매우 상세한 수준에서 대상들은 개별적인 자료마당으로 될수 있다. 레코드, 파일, 지어 전체 자료기지와 같이 매우 집합된 그룹들도 행렬에서 대상으로 될수 있다. 행렬에서 매 항목은 대상에 대한 주동체의 접근권을 가리킨다.

현실에서 접근행렬은 보통 드물게 쓰이며 두 방법들중 한가지 분해방법으로 실현된다. 행렬이 행으로 분해되면 **접근조종목록**들이 얻어 진다(그림 16-3의 ㄴ). 즉 매 대상에 대해서 접근조종목록은 사용자들과 그들의 허용된 접근권을 렬거한다. 접근조종목록은 음적 혹은 공개적인 항목을 포함한다. 이것은 특권을 가진것으로 명백히 렬거되지 않은 사용자들이 음적권한모임을 가지게 한다. 목록의 요소들에는 사용자들의 그룹이나 개별적인 사용자들이 속할수 있다.

렬에 의한 분해로 **자격증**들을 만든다(그림 16-3의 ㄷ). 자격증은 권한이 있는 객체들과 사용자에 대한 조작들을 서술한다. 매 사용자는 표번호를 가지며 그것을 다른 사람에

게 빌려 주거나 줄수 있는 권한을 가질수 있다. 표들은 체계주위에 분산될수 있으므로 그것들은 접근조종목록보다 더 큰 보안문제를 야기시킨다. 특히 표는 위조할수 없는것이여야 한다. 위조하지 못하게 하는 한가지 방도는 사용자들의 이름으로 모든 표들을 장악한 조작체계를 가지는것이다. 이 표들은 사용자들이 접근하기 힘든 기억령역에 있어야 한다.

신용체계의 개념

우리가 지금까지 론의한 많은것은 사용자에 의한 피동 혹은 능동적공격으로부터 주어진 통보문이나 항목을 보호하는것과 련관되여 있다. 조금 다르지만 널리 응용할수 있는 요구는 보안준위에 기초하여 자료나 자원을 보호하는것이다. 이것은 보통 군사분야에서 찾아 볼수 있는데 여기서는 정보를 비밀이 아닌것(U),신용이 있는것(C),비밀인것(S),최고비밀인것(TS) 기타로 분류한다. 이 개념은 다른 령역에 똑같이 응용할수 있는데 여기서는 정보를 크게 몇가지 부류로 나누고 사용자들에게 일정한 부류의 자료의 접근에 대한 통과허가를 줄수 있다. 례를 들어 가장 높은 보안수준은 회사관리들과 부원들만이 접근할수 있는 전략적준위의 중요한 회사의 계획문서와 자료가 될수 있다. 다음 개별적인 관리인,회사직원 등만이 접근할수 있는 중요한 재정자료와 개인자료가 될수 있다.

자료의 다중분류나 준위를 정의할 때의 보안은 **다중준위보안**에 귀착된다. 다중준위보안에 대한 일반적인 한가지 요구는 높은 준위의 주동체는 권한이 있는 사용자의 의사를 정확히 반영하지 않는 한 준위가 낮거나 비교되지 않는 주동체에로 정보를 전달할수 없다는것이다. 실현목적에서 이 요구는 두가지 부분으로 되여 있으며 간단히 규정된다. 다중준위안전체계는 다음의 규칙을 실시하여야 한다.

- **이상 읽지 못한다.** (No read up):주동체는 보안준위가 낮거나 같은 대상을 읽을수만 있다. 이것은 **단순보안속성**의 연구에 귀착된다.
- **이하 쓰지 못한다.** (No write down):주동체는 보안준위가 높거나 같은 대상에만 쓸수 있다. 이것은 문헌들에서 ***-속성**으로 표시한다.

이 두가지 규칙을 정확히 실시하면 다중준위보안을 실현할수 있다. 자료처리체계에서 취급하였으며 많은 연구와 개발의 대상이였던 방식은 **참조감시기개념**에 기초한것이다. 이 방식을 그림 16-4에 제시하였다. 참조감시기는 콤퓨터의 하드웨어와 조작체계에서 주동체와 객체의 보안파라메터들에 기초하여 대상에 대한 주동체의 접근을 규정하는 조종요소이다. 참조감시기는 매 객체의 보호속성(등급준위)과 매 주동체의 접근특권(보안통과허가)을 렬거하는 **보안핵심부자료기지**파일에 접근한다. 참조감시기는 보안규칙들(no read up,no write down)을 실시하며 다음의 속성들을 가진다.

- **완전조정:**보안규칙들을 모든 접근에 대해 실시하지만 실례를 들어 파일을 열 때와 같이 꼭 그런것만은 아니다.
- **고립:**참조감시기와 자료기지는 권한이 없는 변경으로부터 보호된다.
- **검증능력:**참조감시기의 정확성은 증명할수 있어야 한다. 이것은 참조감시기가 보안규칙들을 실시하며 완전조정과 고립을 제공한다는것을 수학적으로 보여 줄수 있어야 한다는것이다.

이것은 강한 요구들이다. 완전조정에 대한 요구는 주기억안과 그리고 디스크와 테프상에 있는 자료에 대한 모든 접근을 조정하여야 한다는것을 의미한다. 순수 쏘프트웨어로 실현하는것은 실제적으로 지나치게 많은 수행부담을 준다. 해결책은 하드웨어에서 적

어도 일부분을 담당하여야 한다는것이다. 고립에 대한 요구는 공격자가 아무리 재간이 있다고 해도 참조감시기의 론리나 보안핵심부자료기지의 내용을 변화시킬수 없게 한다는 것을 의미한다. 이와 같은 검증을 제공할수 있는 체계를 **신용체계**라고 한다.

그림 16-4에 보여 준 마지막요소는 검열파일이다. 검출된 보안위반과 그리고 보안핵심부자료기지에 대한 권한이 부여된 변화와 같은 중요한 보안사건들은 검열파일에 기억된다. 1981년에 미국방성은 자기자신의 요구와 민간봉사를 만족시키기 위해 국가보안국(NSA)내에 신용콤퓨터체계의 폭 넓은 리용을 장려할 목적으로 콤퓨터보안쎈터를 설립하였다. 그 목적은 쎈터의 상품평가프로그람에 의해 실현된다. 본질에 있어서 쎈터의 의도는 보안요구를 만족하는 상업상 유용한 제품을 평가하는것이다. 쎈터는 평가된 상품들을 그것들이 제공하는 보안특징의 범위에 따라 분류한다. 이러한 평가들은 국방성에 필요하였지만 공개되여 자유롭게 쓰이고 있다. 이때부터 그것들은 상업상 유용한 즉 기성설비를 구입하는데서 상업고객들의 안내로서 봉사할수 있었다.

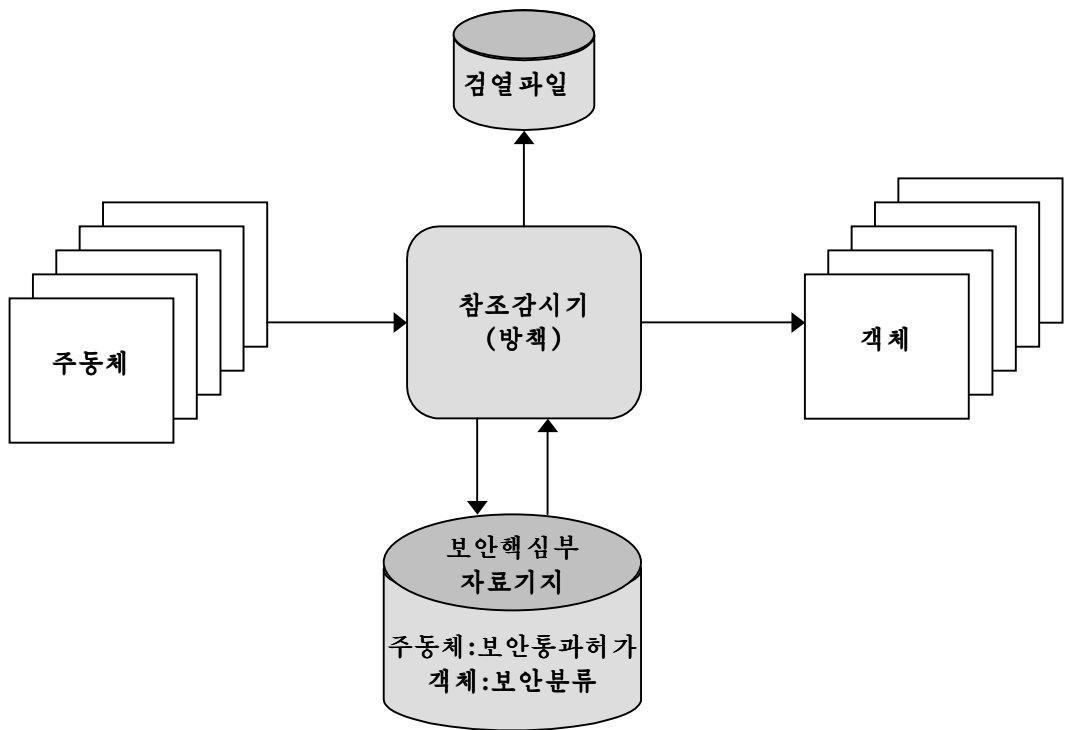



그림 16-4. 참조감시기의 개념

트로이목마방어

트로이목마에 대해 안전한 한가지 방법은 안전하고 믿음직한 조작체계를 리용하는것이다. 그림 16-5에 실례를 보여 주었다. 이 경우에 트로이목마는 대부분의 파일관리와 조작체계가 리용하는 표준보안기구 즉 접근조종목록을 우회하는데 리용한다.

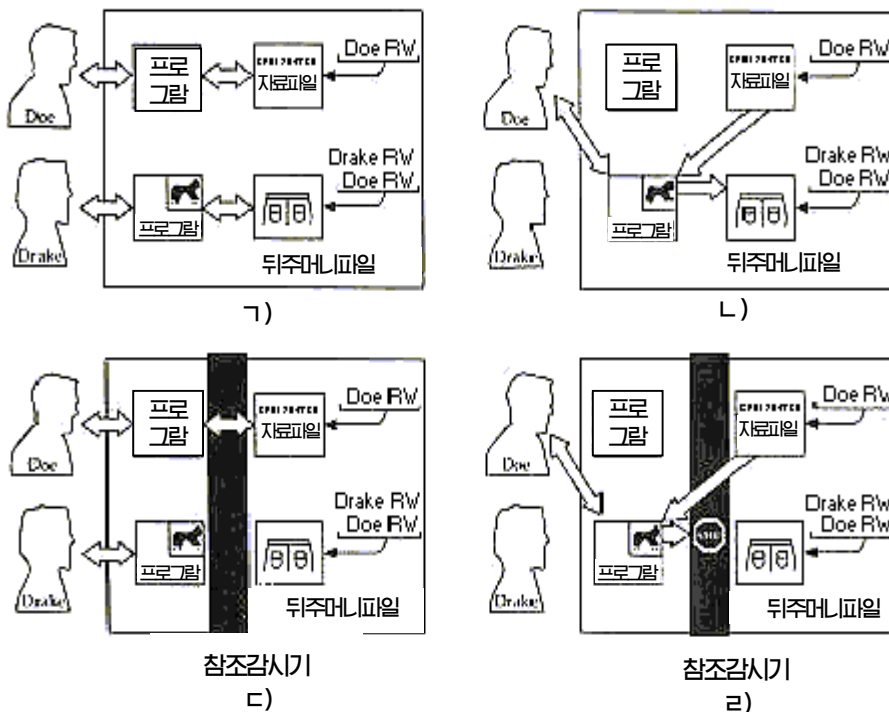



그림 16-5. 트로이목마와 안전한 조작체계

이 실례에서 Doe라고 부르는 사용자는 절대비밀인 기호렬 “CPE1704TKS”을 포함하는 자료파일을 가진 프로그람을 통해 대화한다. 사용자 Doe는 자신을 위해서 실행되는 프로그람에만 제공되는 읽기/쓰기허가를 가진 파일을 만든다. 즉 이것은 Doe가 소유한 처리들만이 파일에 접근할수 있다는것이다.

Drake라고 부르는 적측사용자가 체계에 대한 합법적인 접근을 하려 할 때 트로이목마공격이 시작되며 이것은 트로이목마프로그람과 “뒤주머니(back pocket)”로서 공격에 리용되는 비공개파일을 둘다 설치한다. Drake는 이 파일에 대해 자기자신에게는 읽기/쓰기허가를 주며 Doe에게는 쓰기허가만을 준다(그림 16-5의 ㄱ). Drake는 이제는 이파일이 가치 있는 응용프로그람이라고 광고함으로써 Doe가 트로이목마프로그람을 호출하도록 유도한다. 트로이목마프로그람은 Doe가 실행하고 있다는것을 검출하였을 때 Doe의 파일로부터 비밀기호렬을 복사하고 그것을 Drake의 back-pocket파일에 복사한다(그림 16-5의 ㄴ). 읽기/쓰기의 두 연산은 접근조종목록의 제약조건을 만족시킨다. 다음 Drake는 기호렬의 값을 알기 위해 그후에 자기의 파일에 접근만 하면 된다.

이제 이 씨나리오에서 안전한 조작체계를 리용한다고 하자(그림 16-5의 ㄷ). 보안준위들은 콤퓨터가 접근되여 있고 사용자가 통과암호/ID에 의해 식별되는것으로 포함되는 말단과 같은 기준에 기초하여 가입할 때에 주동체에 할당된다. 이 실례에는 두가지 보안준위 즉 비밀은 공개보다 더 높은 준위에 있도록 순서화한 비밀과 공개가 있다. 처리공정들은 Doe가 소유하며 Doe의 자료파일에 비밀보안준위가 할당된다. Drake의 파일과 처리공정들은 공개보안준위로 국한된다. Doe가 트로이목마프로그람(그림 16-5의 ㄹ)을 호출하면 트로이목마프로그람은 Doe의 보안준위를 얻는다. 따라서 이 프로그람은 단순

한 보안속성하에서 비밀기호렬을 얻을수 있다. 그러나 프로그람이 공개파일(back-pocket파일)에 기호렬을 기억시키려고 할 때 *-속성에 위반되며 이러한 시도는 참조감시기에 의해 허가되지 않는다. 그러므로 back-pocket파일에 쓰려는 시도는 접근조종목록이 그것을 허가하였다 할지라도 거부된다. 즉 보안방책은 접근조종목록기구보다 우선권을 가진다.

참고문헌

앞으로도 매우 읽을 가치가 있다고 볼수 있는 방화벽에 대한 두가지 고전론문들은 문헌 [CHES94]와 문헌 [CHAP95]이다. 문헌 [LODI98],[OPPL97], [BELL94]는 주동체에 대한 좋은 개요론문이다.

문헌 [GASS88]에는 신용콤퓨터체계에 대한 폭 넓은 연구결과들이 제시되였다. 다른 좋은 정보원천은 문헌 [PFLE97]이다. 이 문제에 대한 일부 기초론문들의 재판은 문헌 [ABRA87]에 있다.

ABRA87 Abrams,M. ,and Podell,H. *Computer and Network Security*. Los Alamitos,CA:IEEE Computer Society Press,1987.

BELL94 Bellovin, S., and Cheswick, W. "Network Firewalls." *IEEE Communications Magazine*,September 1994.

CHAP95 Chapman,D. ,and Zwicky,E. *Building Internet Firewalls*. Sebastopol, CA:O'Reilly,1995.

CHES94 Cheswick,W. ,and Bellovin,S. *Firewalls and Internet Security:Repelling the Wily Hacker*. Reading,MA:Addison-Wesely,1994.

GASS88 Grasser,M. *Building a Secure Computer System*. New York:Van Nostrand Reinhold,1988.

LODI98 Lodin,S. ,and Schuba,C. "Firewalls Fend Off Invasions from the Net". *IEEE Spectrum*,February 1998.

OPPL97 Oppliger,R. · "Internet Security:Firewalls and Beyond." *Communications of the ACM*,May 1997.

PFLE97 Pfleeger,C. *Security in Computing*. Upper Saddle River,NJ:Prentice Hall,1997.

문 제

1. 다중준위안전체계에서 《no read up》규칙의 필요성은 매우 명백하다. 《no write down》규칙의 중요성은 무엇인가?
2. 그림 16-5에서 트로이목마복사-관찰사슬의 한개 고리가 끊어 졌다. Drake에 의한 가능한 두가지 공격방법이 있다. 즉 Drake는 가입하여 직접 기호렬을 읽으려고 하며 back-pocket파일에 비밀보안준위를 할당한다. 참조감시기는 이 공격을 막을수 있겠는가?

부록 16: 암호학과 망보안의 강의를 위한 실습과제

많은 교원들은 실습과제들의 실현이 암호학과 망보안을 완전히 리해하는데서 매우 중요하다고 보고 있다. 실습과제들이 없이는 학생들이 구성부분들중의 일부 기본개념들과 호상작용들을 원만히 파악할수 있다. 실습과제들은 책에서 나오는 개념들을 더 보충하고 학생들에게 암호학적알고리듬과 작업규약들이 어떤것인가에 대한 폭 넓은 인식을 주며 학생들이 취미를 가지게도 하고 그들이 보안능력을 구체적으로 리해할수 있을뿐아니라 실현할수도 있다는 자신심을 가지게 한다.

이 책에서 저자는 암호학과 망보안에 대한 개념들을 될수록 구체화하기 위해 약 140개의 숙제문제들을 제시하였다. 그러나 적지 않은 교원들이 실습과제로서 그 자료를 보충할것이라고 본다. 부록에서는 이러한 관점에서 간단한 안내를 주고 교수안을 작성하는데 도움으로 될수 있는 자료들을 주었다. 그러한 자료는 다음과 같은 3가지 실습과제로 되여 있다.

- 연구실습과제
- 프로그람작성실습과제
- 읽기/보고서실습과제

1. 연구실습과제

과목에서 배운 기본개념들을 구체화하고 학생들에게 연구방법을 가르치는 효과적인 방법은 연구실습과제를 선정하는것이다. 이와 같은 실습과제는 인터네트를 통해 상품판매자를 조사하는것과 같은 문헌조사,실험연구,표준화로 되여 있다. 실습과제들은 소조별로 줄수도 있고 더 작은 실습과제들은 개별적으로 줄수도 있다. 어쨌든 교원들이 적당한 제목과 시간수에 대한 계획을 세울수 있도록 시간을 보장함으로써 학기초에 실습과제안을 분류하도록 하는것이 제일 좋다. 학생들에게 다음과 같은 연구실습과제들을 줄수 있다.

- 계획에 대한 양식
- 마지막보고서에 대한 양식
- 중간과 마감경계에 대한 계획
- 가능한 대상과제제목들의 목록

대학생들은 렬거된 제목들에서 어느 하나를 선택하거나 그것들과 류사한 실습과제를 생각해 낼수 있다. 교원용참고서에는 15개의 가능한 연구제목의 목록은 물론 계획과 마지막보고서에 대한 양식이 들어 있다.

2. 프로그람작성실습과제

프로그람작성실습과제는 효과적인 교육방법이다. 현존 보안기교의 부분으로 되지 않는 단독프로그람작성실습과제들에는 여러가지 흥미 있는 특징들이 있다.

1. 교원은 실습과제를 암호학과 망보안개념의 넓은 분야에서 선정할수 있다.
2. 실습과제들은 학생들이 임의의 콤퓨터에서 임의의 적합한 언어로 프로그람을 작성할수 있게 한다. 즉 실습과제들은 가동환경이나 언어에 무관계하다.
3. 교원은 실습대상과제에 대한 임의의 개별적인 하부조직을 설치하고 구성할 필요가 없다.

실습과제들의 크기도 각이할수 있다. 보다 큰 실습과제들은 대학생들에게 큰 포부를 주지만 능력이 낮거나 조직적수완이 부족한 학생들은 뒤떨어 질수 있다. 보다 큰 대상과제들은 보통 우수한 학생들이 진지한 노력을 발휘하게 한다. 가장 작은 실습과제들에서는 개념 대 코드의 비률이 더 높고 그것들중 많은것들을 선정할수 있으므로 서로 다른 령역들을 취급할수 있다.

한편 연구실습과제와 마찬가지로 대학생들은 처음에 계획을 제출해야 한다. 대학생들의 견본품은 1에서 서술한것과 같은 요소들을 포함할것이다. 교원용참고서에는 12개의 가능한 프로그람작성대상과제들이 들어 있다.

교원용참고서에 제안한 연구와 프로그람작성대상과제들을 보충한 사람들은 꼴롬비아종합대학의 Henning Schulzrinne, 오렌곤주종합대학의 Cetin Kaya Koc와 그리고 신용정보체계와 죠지워싱톤종합대학의 David M. Balenson이다.

3. 읽기/보고서실습과제

과정안의 개념을 구체화하며 대학생들에게 연구경험을 주는 다른 좋은 방법은 문헌들에서 읽고 분석하여야 할 론문들을 선정하는것이다. 다음과 같은 선정이 있을수 있다.

연구문헌에서 론문을 읽고 보고서를 작성한다. 보고서는 한두페지로 작성할수 있다. 보고서내용의 4분의 3은 론문에 대한 개요이고 나머지가 평론부분으로 된다. 교과서의 《참고문헌》부분에서와 같은 양식을 리용하여 보고서에 론문의 형식적인용방법을 적용한다.

교수안에는 지정된 론문들에 대한 목록란이 들어 있다. 모든 론문들은 인터네트를 통해서든가 임의의 해당한 대학과학도서관에서 쉽게 볼수 있다. 도서의 매장마다 론문들이 소개되여 있다.

용어해설

강제적접근조종(Mandatory Access Control)
사용자들 그리고 파일들과 다른 대상들에 할당된 고정된 보안속성들에 기초하여 대상에 대한 접근을 제한하는 방법이다. 조종을 사용자나 그들의 프로그람에 의해 변경할수 없다는 의미에서 위임이다.

공개열쇠(Public Key)
비대칭암호체계에서 리용되는 두 열쇠중의 한 열쇠이다. 공개열쇠는 대응하는 비공개열쇠와의 련결에 리용하기 위해 공개된다.

공개열쇠암호(Public Key Encipher)
비대칭암호

계산학적으로 안전한(Computationally Secure)
보안을 파괴하는 비용과 시간을 and/or한것이 실행할수 없을 정도로 너무 커서 안전한것을 말한다.

도식(Diagram)
두 글자렬이다. 영어와 다른 언어로 작성된 평문에 대한 여러 도식의 상대적인 빈도수는 일부 암호의 암호분석에 리용할수 있다. Diagraph(분도자)라고도 부른다.

대칭암호(Symmetric Encryption)
암호화와 복호화를 같은 열쇠를 리용하여 진행하는 암호형식이다. 전통암호라고도 한다.

대화열쇠(Session Key)
두 사람사이에 리용하는 림시암호화열쇠이다.

론리폭탄(Logic Bomb)
론리는 체계에 나타나야 할 일정한 조건모임에 대해 검사하는 콤퓨터프로그람의 한 부분으로 되여 있다. 이 조건들을 만족할 때 부당한 동작들의 결과인 어떤 기능이 실행된다.

모조란수생성기(Pseudorandom Number Generator)
겉보기통계적우연수인 수렬을 결정론적으로 만드는 함수이다.

무조건적으로 안전한 비밀(Unconditionally Secure)
아무리 무한한 시간과 무한한 콤퓨터자원을 가진 적수에 대하여서도 안전한 비밀이다.

박테리아(Bacteria)
자기자신의 복제로 하여 체계자원을 소비하는 프로그람이다.

변환통로(Covert Channel)
통신시설의 설계자가 의도한것과는 다른 방법으로 정보를 이송할수 있는 통신로이다.

복호화(Decryption)
암호화된 본문이나 자료(암호문이라고 부른다.)를 본래의 본문이나 자료(평문이라고 부른다.)로 변환하는것이다. 분석이라고도 부른다.

부호(Code)
정보의 일부(즉 글자, 단어, 성구)를 반드시 같은 부류의것이 아닌 다른 대상으로 치환하는 일정한 규칙이다. 일반적으로 의미를 숨기려고 하는것은 아니다. 실례로 ASCII(매 기호는 7bit로 표현된다.)와 빈도수-밀기열쇠(매 2진값은 개별적인 빈도수를 나타낸다.)를 들수 있다.

블로크련쇄(Block Chaining)
현재의 평문입력블로크와 열쇠뿐아니라 이전의 입력과 출력을 and/or한것을 출력블로크로 하는 대칭블로크암호화에서 리용하는 수속이다. 블로크사슬의 효과는 두개의 같은 평문입구블로크를 암호분석이 보다 어려운 서로 다른 암호문블로크로 만드는것이다.

블로크암호(Block Cipher)
평문비트들의 큰 블로크(일반적으로 64bit)를 같은 길이의 암호문블로크로 변환하는 대칭암호화알고리듬이다.
비공개열쇠(Private Key)
대칭암호화체계에서 리용하는 두개의 열쇠중 한 열쇠를 말한다. 비밀통신에서 비공개열쇠는 그것을 만든 사람만이 알수 있다.
비대칭암호화(Asymmetric Encryption)
암호화와 복호화에 두개의 서로 다른 열쇠를 리용하는 암호체계형식인데 한 열쇠는 공개열쇠이고 다른 열쇠는 비공개열쇠이다. 공개열쇠암호화라고도 한다.
비루스(Virus)
한개 혹은 그이상의 다른 프로그람에 삽입될수 있도록 자가복제를 생성하는 프로그람안에 들어 있는 코드이다. 비루스는 전염외에 보통 어떤 바람직 하지 않는 기능을 수행한다.
비밀열쇠(Secret Key)
대칭암호체계에서 리용하는 열쇠이다. 쌍방은 같은 열쇠를 가져야 하며 그 열쇠는 안전한 통신을 위해 비밀로 되여야 한다.
사태효과(Avalanche Effect)
평문이나 열쇠의 작은 변화가 암호문의 큰 변화를 일으키게 하는 암호알고리듬의 특성이다. 하쉬코드에서의 사태효과는 통보문의 작은 변화가 통보문개요의 큰 변화를 일으키게 하는 특성이다.
수자서명(Digital Signature)
서명코드를 첨가한 통보문을 만들게 하는 인증기구이다. 서명은 원천지(보낸사람)와 통보문의 완정성을 보증한다.
신용체계(Trusted System)
주어 진 보안방책을 실현한것을 검증할수 있는 콤퓨터와 조작체계이다.
자의적접근조종(Discretionary Access Control)
객체들에 대한 접근을 그들이 속한 주동체나 그룹들의 식별에 기초하여 제한하는 방법이다. 확실한 접근허가를 가지는 주동체는 임의의 다른 주동체(강제적접근조종에 의해 제한되지 않을 때까지)에 대한 허가(아마 간접적으로)를 통과할수 있다는 의미에서 조종들은 임의로 결정된다.
전통암호화(Conventional Encryption)
대칭암호화
주열쇠(Master Key)
열쇠배포쎈터와 웃준위사이의 대화열쇠들의 전송을 암호화할 목적으로 리용하는 매우 긴 열쇠이다. 일반적으로 주열쇠는 암호가 아닌 수단으로 배포된다. 열쇠암호화열쇠라고도 한다.
재연공격(Replay Attacks)
이미 권한이 부여되여 완수된 봉사가 다른 《복제요구》에 의하여 권한이 부여된 지령을 반복하려는 시도에서부터 위조되는 공격이다.
차분암호분석(Differential Cryptoanalysis)
매 XOR차패턴에 의해 선택한 평문을 암호화하는 기술이다. 암호문의 결과인 차패턴은 암호열쇠를 결정하는데 리용할수 있는 정보를 제공한다.
초기화벡토르(Initialization Vector)
블로크련쇄암호화기술을 리용할 때 다중평문블로크를 암호화하기전에 리용하는 우연수자료블로크이다. IV는 평문기초공격을 좌절시키는것으로써 봉사한다.
침입자(Intruder)
콤퓨터체계에 대한 부당한 접근이나 특권을 얻으려고 하는 혹은 얻은 사람이다.
통과암호(Password)
식별자를 인증하기 위해 리용하는 기호렬이다. 통과암호와 그와 관련한 사용자ID의 지식은 이 사용자ID와 관련한 자격들을 리용하기 위한 권한증명이라고 볼수 있다.
통보문개요(Message Digest)
하쉬함수
통보문인증코드(MAC)
암호학적검사합

트로이목마(Trojan Horse)
보안을 파괴하기 위해 처리공정에 접근할수 있는 합법적인 권한을 몰래 리용하는 추가적인(숨은) 기능을 가지는 쓸모있어 보이는 콤퓨터프로그람이다.

평문(Plaintext)
암호화함수의 입력 혹은 복호화함수의 출력이다.

하쉬함수(Hash Function)
변하는 길이를 가지는 자료블로크나 통보문을 하쉬코드라고 부르는 고정된 길이를 가지는 값으로 넘기는 함수이다. 함수는 보호할 때 자료나 통보문에 대한 인증자를 준다는 의미에서 설계한다. 통보문개요라고도 한다.

한방향함수(One Way Function)
함수값계산은 쉽게 할수 있지만 거꿀계산은 할수 없는 함수이다.

한번쓰기정보(Nonce)
한번만 리용되는 식별자나 번호이다.

함정문(Trap Door)
표준적으로 접근을 인증하는 방법이 없이 접근하는데 리용하는 프로그람에로의 나타나지 않은 비밀입구점이다.

함정문한방향함수(Trap-Door One-Way Function)
쉽게 계산할수 있는 함수이다. 이것의 거꿀계산은 정확한 특권정보를 알기전까지는 계산할수 없다.

혼란(Confusion)
암호문의 통계량과 암호열쇠의 값사이의 관계를 될수록 복잡하게 하는 암호기술이다. 이것은 열쇠와 입력값에 관계되는 복잡하고 불규칙적인 알고리듬을 리용하여 달성한다.

흐름암호(Stream Cipher)
평문입력의 흐름으로부터 암호문출력을 비트-비트 혹은 바이트-바이트로 만드는 대칭암호알고리듬이다.

확산(Diffusion)
개개의 개별적인 평문수자가 많은 암호문수자들에 영향을 줌으로써 평문의 통계학적구조를 애매하게 하는 암호학적기술이다.

암호(Cipher)
암호화와 복호화알고리듬이다. 암호는 의미를 숨기려는 목적에서 정보토막(평문의 요소)을 다른 대상으로 치환한다. 일반적으로 치환규칙은 비밀열쇠에 의해 조종된다.

암호문(Ciphertext)
암호알고리듬의 출력이며 통보문이나 자료의 암호화된 형식이다.

암호분석(Cryptanalysis)
정보를 회복하기 위해 암호를 파괴하거나 혹은 인증하여 접수할수 있는 암호화된 정보를 위조하는것과 관련되는 암호학의 한 분야이다.

암호학(Cryptography)
통보문의 기밀성과 신뢰성의 and/or를 믿게 하는 암호화와 복호화알고리듬의 설계와 관련한 암호연구의 한 분야이다.

암호학적검사합(Cryptographic Checksum)
인증하여야 할 자료와 비밀열쇠에 관한 암호학적함수가 있는 인증자이다. 통보문인증코드(MAC)와 관련되여 있다.

암호화(Encryption)
평문이나 자료를 변환표나 알고리듬에 기초하여 거꿀변환방법으로는 알기 힘든 형태로 변환하는 방법이다. Enciphering이라고도 부른다.

암호연구(Cryptology)
암호학과 암호분석학을 둘 다 포함하는 비밀통신에 관한 연구이다.

여러준위보안(Multilevel Security)
분류한 자료의 다중준위를 통과하는 접근조종을 실시하는 능력이다.

여러준위암호화(Multilevel Encryption)
평문을 암호문으로 더 복잡하게 넘기기 위해 서로 다른 열쇠들로써 암호화함수를 반복리용하는 방법이다.

열쇠배포쎈터(Key Distribution Center)
웃준위에 림시대화열쇠를 보내는 권한을 가진 체계이다. 매 대화열쇠는 열쇠배포쎈터가 목적지상급과 공유하는 주열쇠를 리용하여 암호화한 형태로 전송된다.

인증(Authentication)

전송된 자료 특히 통보문의 완정성을 검증하는데 리용되는 방법이다.

인증자(Authenticator)

수신자가 통보문을 믿을만 하다는것을 검증할수 있도록 통보문에 첨가한 추가적인 정보이다. 인증자는 기능적으로 통보문자체의 내용과 독립일수 있거나(즉 한번쓰기정보이거나 원천지식별자) 통보문내용에 관한 함수로 될수 있다(즉 하쉬값이거나 암호학적검사합).

웜(Worm)

자기자신을 복제하여 망접속을 통해 콤퓨터로부터 콤퓨터에로 복사물을 보내는 프로그람이다. 웜은 도착하여 다시 복제되며 전염될수 있다. 전염외에 웜은 보통 어떤 바라지 않는 기능을 수행한다.

Kerberos

프로젝트아쎄나(Project Athena)코드에 인증봉사를 주는 이름이다.

RSA알고리듬(RSA Algorithm)

모드대수의 제곱에 기초한 공개열쇠암호알고리듬이다. 이 알고리듬만이 일반적으로 공개열쇠암호에서 실용적이며 안전하다고 인정되여 있다.

참 고 문 헌

ABRA87 Abrams.M., and Podell,H.*Computer and Network Security*. Los Alamitos, CA:IEEE Computer Society Press,1987.

ABRA95 Abrams,M.;Jajodia,S.;and Podell,H.eds.*Information security:An Integrated Collection of Essays*. Los Alamitos,CA:IEEE Computer Society Press,1995.

ADAM90 Adams,c,and Tavares,S." Generating and Counting Binary Bent sepuences." *IEEE Transactions on Information Theory*,1990.

ADAM92 Adam,J." Virus Threats and Countermeasures." *IEEE spectrum*,August 1992. Simple and Effective Key Scheduling for Symmetric_Ciphers." *Proceedings,Workshop in Selected Areas of Cryptograph,SAC'* 94.1994.

ADAM97a Adams,C.

ADAM94 Adams,C. "Constructing Symmetric Ciphers Using the CAST Design." *Designs,Codes,and Cryptograph*,1997.

AMAM97b Adams,C.The CAST-128 *Encryption Algorithm*.RFC2144,May 1997.

AKL83 Akl,S." Digital Signnatures:A Tutorial Survey." *Computer*,February 1983.

ALV90 Alvare,A." How Crackers Crack Passwords or What Passwords to Avoid." *Proceedings,UNIX Security Workshop* II,August 1990.

ANDE80 Anderson,J. *Computer Security Threat Monitoring and Surveillance*.Fort Washington,PA:James P.Anderson Co.,April 1980.

BALD96 Baldwin,R.,and Rivest,R. *The RC5,RC5-CBC,RC5-CBC-Pad,and RC5-CTS Algorithms*. RFC2040, October 1996.

BARK91 Barker,W.*Introduction to the Analysis of the Data Encryption Standard* (*DES*).Laguna Hills,CA:Aegean Park Press,1991.

BAUE88 Bauer,D.,and Koblentz,M. "NIDX-An Expert System for Real-Time Network Intrusion Detection." *Proceedings,Computer Networking Symposium*,April 1988. "

BELL90 Bellovin,S.,and Merritt,M. "Limitations of the Kerberos Authentication System." *Computer Communications Review*,October 1990.

BELL92 Bellovin,S. "Three Be Dragons." *Proceedings,UNIX Security Symposium III*,September 1992.

BELL93 Bellovin,S. "Packets Found on an Internet." *Computer Communications Review*,July 1993.

BELL94 Bellovin,S.,and Cheswick,W. "Network Firewalls." *IEEE Communications Magazine*,September 1994.

BELL96a Belare,M.;Canetti,R.;and Krawczyk,H." Keying Hash Functions for Message Authentication." *Proceedings,CRYPTO' 96*,August 1996;published by Springer-Verlag. An expanded version is available at http://www-cse.ucsd.edu/users/mihir.

BELL96b Bellare, M. ; Canetti, R. ; and Krawczyk, H. "The HMAC Construction." *CryptoBytes*, Spring 1996.
BELL97 Bellare, M., and Rogaway, P. "Collision-Resistant Hashing: Towards Marking UOWHF' s Practical." *Proceedings, CRYPTO ' 97*, 1997; published by Springer-Verlag.
BERG97 Berg, A. "Viruses: More Infections than Ever, Users Say." *LAN Times*, June 23, 1997.
BERS92 berson, T. "Differential Cryptanalysis Mod 2^{32} with Applications to MD5." *Proceedings, EUROCRYPT ' 92*, May 1992; published by Springer-Verlag.
BETH91 Beth, T. ; Frisch, M. ; and Simmons G. eds. *Public-Key Crytography: State of the Art and Future Directions.* New York: Springer-Verlag, 1991.
BIHA93Biham, E., and Shamir, A. *Differential Cryptanalysis of the Data Encryption Standard*. New York: Springer-verlag, 1993.
VLUM86 Blum, L. ; Blum, M. ; and Shub, M. "A Simple Unpredictable Pseudo-Random Number Generator." *SIAM Journal on Computing*, No. 2, 1986.
BLOO70 Bloom, B." Space/Time Trade-offs in Hash Coding with Allowable Errors." *Communications of the ACM*, July 1970.
BOER93 Boer, B., and Bosselares, A. "Colllisions for the Compression Function of MD%." *Proceedings, EUROCRYPT ' 93*, 1993; published by Springer-Verlag.
BOSS96 BOSSELAERS, A., ; Govarets, R. ; and Vandewlle, J." Fast Hashing on the Pentium" *Proceedings, Crypto ' 96*, August1996; published by Springer-verlag
BOSS97 Bosselaers, A. ; Dobbertin, H; and Preneel, B. "TheRIPEMD-160 Cryptographic Hash Function." *Dr.Dobb ' s Journal*, January 1997.
BOWL92Bowles, J. and Pelaze, C." Bad Code." *IEEE Spectrum*, August 1992.
BRAD96 Brander, s., and Mankin, A. *Ipng: Internet protocol Next Generation* Reading, MA: Addition-Wesley, 1996.
BRIG79 Bright, H., and Ension, R. "Quasi-Random Number Seqences fromLong-Period TLP Generatorwith Remarks on Applicationto Crytography." *Computing surveys*, December 1979.
Brya88 Bryant, W. *Designing an Autentication System*: dialogue in Four Scenne s. Project Athena document, Feburary1988. Available at http://web.mit.edu/kerberos/www/dialogue.html.
BURN97 Burn, R. a *Pathway to Number Theory*. Cambridge, England: Cambridge University Press, 1997
CAMP92 Campbell, K., and Wiener, M. "Proof that DESaIs Not a Group." *Proceedings, Crypto ' 92*, 1992; published by Springer-verlag.
CHAP95 Chapman, D., and Zwicky, *E., Building Internet Firewalls*. Sebastopo l, CA; O' Reilly, 1995.
CHEN98 Cheng, P., et al. "A Security Architecture for the Internet Protocol." *IBM Systems Journal*, Number 1, 1998.
CHES94 Cheswick, W., and Bellovin, S. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, MA: Addison-Wesely, 1994.

CHES97 Chess, D. "The Future of Viruses on the Internet." *Proceedings, Virus Bulletin International Confrence,*Octerber 1997.
COCK73 Cocks, C. A *Note on Non-Secret Encrytion*. CESG Report, November 1973.
COHE94 Cohen, F.*A Short Course on Comput*er *Viruses*. New York: Wiley, 1994.
COME95 Comer, D. *Internetworking with TCP/IP, Volume I: Principles, Protocols and Architecture*. Upper Saddle River, NJ: Prentice Hall, 1995.
COOP89 Cooper, J. *Computer and Communications Security: Strategies for the 1990s*. New York: McGraw-Hill, 1989.
COPP94 Coppersmith, D. "The Data Encryption Standard (DES) and Its Strngth Against Attacks." *IBM Journal of Research and Development*.
CORM90 Cormen, T.; Leiserson, C.; and Riverst, R. *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.
COWE96 Cowie, J., et al. "A World Wide Number Field Sieve Factoring Record: On to 512 Bits." *Proceedings, ASIACRYPT' 96*, November, 1996; published by Springer-Verlag.
DAMG89 Damgard, I. "A Design Principle for Hash Functions." *Proceedings, CRYPTO' 89*, 1989; published by Springer-Verlag.
DAVI89 Davies, D., and Price, W. *Security for Computer Network*. New York: Wiely, 1989
DAVI93 Davies, D., and Ganesan, R." Bapasswd: A New Proactive Password Checker." *Proceedings, 16th National Computer Security Conference*, September 1993
DENN81 Denning, D." Timestamps in Key Distribution Protocols." *Communications of the ACM*, August 1981.
DENN82 Denning, D. *Cryptography and Data Security*. Reading, MA: Addison-Wesley, 1982.
DENN83 Denning, D. "Protecting Public Keys and Signature Keys." *Computer*, February 1983.
DENN87 Denning, D. "An Intrrusion-Detection Model." *IEEE Transactions on Software Enginering,*February 1987.
DENN90 Denning, P. *Computers Under Attack: Intruders, Worms, and Viruses*. Reading, MA: Addison-Wesley, 1990.
DIFF76a Diffe, W., and Hellman, M. "Multiuser Cryptoraphic Tecniques." *Proceedings of the AFIPS National Computer Conference,*June 1976.
DIFF76b Diffie, W., and Hellman, M. "New Direction in Cryptography." *IEEE Transactions on Information Theory*, November 1976.
DIFF77 Diffie, W., and Hellman, M. "Exhaustive Cryptanalysis of the NBS Data Encryption Standard." *Computer*, June 1977.
DIFF79 Diffie, W., and Hellman, M. "Privacy and Authentication: An Introduction to Cryptography." *Proceedings of the IEEE*, March 1979.
DIFF88 Diffie, W. "The First Ten Years of Public-Key Cryptography." *Proceedings of the IEEE*, May 1988. Reprinted in [SIMM92a].
DOBB96a Dobbertin, H. "The Status of MD5 After a Recent Attack." *Crypto-Bytes*, Summer 1996.

DOBB96b Dobbertin, H, ; Bosselaers, A. ; and Preneel, B. "RIPEMD-160: A Stned Version of RIPEMD." *Proceedings, Third International Workshop on Fast Software Encrytion*, 1996; published by Springer-Verlag.

EAST94 Eastlake, D. ; Crocker, S. ; and Schiller, J. *Randomness Recommerndations for Security*. RFC 1750, December 1994.

ELGA85 Elgamal, T. "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." *IEEE Transactions on Information Theory*, July 1985.

ELLI70 Ellis, J. The *Possibility of Secure Non-Secret Digital Encryption*. CESG Report, January 1970.

ELLI87 Ellis, J. *The* Story *of Non-Secret Encryption*. CESG Report, 1987. Available at http://www.cesg.gov.uk/ellisint.htm

ENGE80 Enger, N., and Howerton, P. *Computer Security*. New York: Amacom, 1980.

FEIS73 Feistel, H. "Cryptography and Computer Privacy." *Scientific American*, May 1973.

FEIS75 Feistel, H. ; Notz, W. ; and Smith, J. "Some Cryptographic Tecniques for Machine-to-Machine Data Communications." *Proceedings of the IEEE*, novvember 1975.

FORD95 Ford, W. "Advances in Public-Key Certificate Standards." *ACM SIGSAC Review*, July 1995.

FORR97 Forrest, S. ; Hofmeyr, S. ; and Somayaji, A. "Computer Immunology." *Communications of the ACM*, Octeber 1997.

FREE93 Freedman, D. "The Goods on Hacker Hoods." *Forbes ASAP*, 13 September 1993.

FUMY93 Fumy, S., and Landrock, P. "Principles of Key Management." *IEEE Journal on Selected Areas in Communications*, June 1993.

GARD72 Gardner, M. *Codes, Ciphers, and Secret Writing*. New York: Dover, 1972.

GARD77 Gardner, M. "A New Kind of Cipher That Would Take Millions of Years to Break." *Scientific American*, August 1977.

GARF97 Garfinkel, S., and Spafford, G. *Web Security & Commerce*. Cambridge, MA: O' Reilly and Associates, 1997.

GASS88 Gasser, M. *Building a Secure Computer System*. New York: Van Nostrand Reinhold, 1988.

GONG92 Gong, L. "A Security Risk of Depending on Synchronized Clocks." *Operating Systems Review*, January 1992.

GONG93 Gong, L. "Variations on the Themes of Message Freshness and Replay." *Proceedings, IEEE Computer Security Foundations Workshop*, June 1993.

GRAH94 Graham, R. ; Knuth, D. ; and Patashnik, O. *Concrete Mathematics: A Foundation for Computer Science*. Reading, MA: Addison-Wesley, 1994.

HAMM91 Hamming, R. *The Art of Probability for Scientists and Engineers*. Reading, MA: Addison-Wesley, 1991.

HEBE92 Heberlein, L.; Mukherjee, B.; and Levitt, K. "Internetwork Security Monitor" An Intrusion-Detection System for Large-Scale

Networks." *Proceedings, 15th National Computer Security Conference*, October 1992.
HELD96 Held, G. *Data and Image Compression: Tools and Techniques*. New York: Wiley, 1996.
HEYS95 Heys, H., and Tavares, S. "Avalanche Characteristics of Substitution-Permutation Encrytion Networks." *IEEE Transactions on Computers*, September 1995
HOFF90 Hoffman, L., editor. *Rogue Programs: Viruses, Worms, and Trojan Horses*. New York: Van Nostrand *Reinhold*, 1990.
HUIT98 Huitema, C. Ipv6: *The New Internet Protocol*. Upper Saddle River, NJ: Prentice Hall, 1998.
IANS90 I'Anson, C., and Mitchell, C. "Security Defects in CCITT Recommendation X.509-The Dictionary Authentication Framework." *Computer Communications* Review, April 1990.
ILGU93 Ilgun, K. "USTAT: A Real-Time Intrusion Detection System for UNIX." *Proceedings, 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, May 1993.
JAIN91 Jain, R. *The Art of Compute Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York: Wiley, 1991.
JAVI91 Javitz, H., and Valdes, A. "The SRI IDES Statistical Anomaly Detector." Proceedings, 1991 *IEEE Computer Society Symposium on Research in Security and Privacy*, May 1991
JOHN97 Johnson, N. *Steganography*. Available at http://patriot.net/~johnson/html/neil/stegdoc.html. 1997
JONE82 Jones, R. "Some Techniques for Handling Encipherment Keys." *ICL Technical Journal*, November 1982.
JUNE85 Jueneman, R.; Matyas, S.; and Meyer, C. "Message Authentication." *IEEECommunication Magazine*, September 1985.
JUNE87 Juneneman, R. "Electronic Document Authentication." *IEEE Network Magazine*, April 1987.
JURI97 Jurisic, A., and Menezes, A. "Elliptic Curvers and Cryptography." *Dr.Dobb's Journal, April 1997.*
KAHN96 Kahn, D. *The Codebreakers: The Story of Secret Writing*. Nesw York: Scribner, 1996.
KALI95 Kaliski, B., and Robshaw, M. "The Secure Use of RSA." *CryptoBytes*, Autumn 1995.
KALI96 Kaliski, B., and Robshaw, M. "Multiple Encryption: Weighing Security and Performance." *Dr. Dobb's Journal*, January 1996.
KEHN92 Kehne, A.; Schonwalder, J.; and Langendorfer, H. "A Nonce-Based Protocol for Multiple Authentications" *Operating Systems Review*, October 1992.
KENT77 Kent, S. "Encryption-Based Protection for Interactive User/Computer Communication." *Proceeding of the Fifth Data Communications Symposium*, September 1977.

KENT93a Kent, S. "Architectual Security," in [LYNC93].
KEPH97a Kephart, J.; Sorkin, G.; Chess, D.; and White, S. "Fighting Computer Viruses." *Scientific American*, November 1997.
KEPH97b Kephart, J.; Sorkin, G.; Swimmer, B.; and White, S. "Blueprint for a Computer Immune System." *Proceeding, Virus Bulletin InternationalConfference*, October 1997.
KLEI90 Klein, D. "Foiling the Cracker: A Survey of, and Improvenments to, Password Security." *Proceeding, UNIX Security Workshop II*, August 1990.
KNUT97 Knuth, D. *The* of *Art Computer Programming, Volume 1: Fundamental Algorithms*. Reading, MA: Addison-Wesley, 1997.
KNUT98 Knuth, D. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Reading, MA: Addison-Wesley, 1998.
KOBL92 Koblas, D., and Koblas, M. "SOCKS." *Proceedings, UNIX Security Symposium III*, September 1992.
KOBL94 Koblitz, N. *A Course in Number Theory and Cryptography*. New York: Springer-Verlag, 1994.
KOCH96 Kocher, P. "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems." *Proceedings, Crypto '96*, August 1996; published by Springer-Verlag.
KOHL89 Kohl, J. "The Use of Encryption in Kerberos for Network Authenticcation." *Proceedings, Crypto '89*, 1989; published by Springer-Verlag.
KOHL94 Kohl, J.; Neuman, B,; and Ts'o, T. "The Evoluation of the Kerberos Authentication Service." In Brazier, F., and Johansen, D. *Distributed Open Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1994. Available at http://web.mit.edu/kerberos/www/papers.html.
KOHN78 Kohnfelder, L. *Towards a Practical public-Key Cryptosystem. Bachelor's* Thesis, M.I.T., May 1978.
KONH81 Konheim, A. *Cryptography: A Primer*. New York: Wiley, 1981.
KORN96 Korner, T. *The Pleasures of Counting*. Cambridge, England: Cambridge University Press, 1996.
KUMA98 Kumanduri, R., and Romero, C. *Number Theory with Computer Applications*. Upper Scaddle River, NJ: Prentice Hall, 1998.
LAI90 Lai, X., and Massey, J. "A Proposal for a New Block Encryption Standard." *Proceeding, EUROCRYPT '90*, 1990; published by Springer-Verlag.
LAI91 Lai, X., and Massey, J. "Markov Ciphers and Differential Cryptanalysis." *Proceedings, EUROCRYPT '91*, 1991; published by Springer-Verlag.
LAI92 Lai, X. *On the Design and Security of Block Ciphers*. Konstanz, Germany: Hartung-Gorre, 1992.
LAM92a Lam, K., and Gollmann, D. "Freshness Assurance of Authentication Protocols." *Proceedings, ESORICS '92*, 1992; published by Springer-Verlag.
LAM92b Lam, K., and Beth, T. "Timely Authentication in Distributed Systems." *Proceedings, ESORICS '92*, 1992; published by Springer-Verlag.

LE93 Le, A.; Matyas. S.; Johnson, D.; and Wilkins, J. "A Public Key Extension to the Common Cryptographic Ahchitecture." *IBM Systems Journal*, No. 3. 1993.
LEHM51 Lehmer, D. "Mathematical Methods in Large-Scale Computing." *Proceedings, 2nd Symposium on Large-Scale Digital Calculating Machinery*, Camdridge: Harvard University Press, 1951.
LEVE90 Leveque, W. *Elementary Theory of Numbers*. New York: Dover, 1990.
LEWI69 Lemis, P.; Goodman, A.; and Miller, J. "A Pseudo-Random Number Generator for the System/360." *IBM Systems Journal*, No. 2. 1969.
LODI98 Lodin, S., and Schuba, C. "Firewalls Fend Off Invasions from the Net." *IEEE Spectrum*, February 1998.
LUNT88 Lunt, T., and Jagannathan, R. "A Prototype Real-Time Intrusion-Detection Expert System." *Proceedings, 1988 IEEE Computer Society Symposium on Research in Security and Privacy*, April 1988.
LYNC93 Lynch, D., and Rose, M., editors. *Internet System Handbook*. Reading, MA: Addison-Wesley, 1993.
MACG97 Macgregor, R.; Ezvan, C.; Ligurori, L.; and Han, J. *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice*. IBM RedBook SG24-4978-00, 1997. Available at www.redbooks.ibm.com/SG244978.
MADS93 Madsen, J. "World Record in Password Checking." *Usenet, comp.security.misc newgroup*, August 18, 1993.
MARK97 Markham, T. "Internet Security Protocol." *Dr. Dobb's Journal*, June 1997.
MATS93 Matsui, M. "Linear Cryptanalysis Method forz DES Cipher. *Proceedings, EUROCRYPT' 93*, 1993; published by Springer-Verlag York: Wiley, 1982.
MATY91a Matyas, S. "Key Handling with control Vectors." *IBM Systems journal*, No. 2, 1991.
MATY91b Matyas, S.; Le, A.; and Abraham, D. "A Key-Management Scheme Based on Control Vectors." *IBM Systems Journal*, No. 2. 1991.
MENE93 Manezes, A. *Elliptic Curve Public Key Cryptosystems*. Boston: Kluwer Academic Publishers, 1993.
MENE97 Menezes, A ; Oorshcot, P.; and Vanstone, S. *Handbook of Applied Crytography*. Boca raton, FL: CRC Press, 1997
MERK79 Merkle, R. *Secrecy, Authentication, and Public Key Systems*. Ph. D. Thesis, Stanford University, June 1979.
MERK81 Merkle, R., and Hellman, M. "0n the Security of Multiple Encryption." *Communictions of the ACM*, July 1981.
MERK89 Merkle, R. "0ne Way Hash Functions and DES." *Proceedings, CRYPTO' 89*, 1989; published by Springer-Verlag.
MEYE82 Meyer, C., and Matyas, S. Cryptography: *A New dimension in computer Data Security*. New York: Wiley, 1982
MEYE88 Meyer, C., and Schilling, M. "Secure Program Load with Modification Detection Code." *Proceedings, SECURICOM' 88*, 1988.

MILL75 Miller,G. “Riemann’s Hypothesis and Tests for Primality.” *Proceedings* of se*venth Annual ACM Symposium on the Theory of Computing* ,May 1975
MILL88 Miller,S.;Neuman,B.,Schiller,J.;and Saltzer,J. “Kerberos Authentication and Authorization System.” *Section E.2.1,Project Athena Technical Plan*,M.I.T.Project Athena,Cambridge,MA.27 October 1988.
MIST96 Mister,S.,and Adams,C. “Practical S-Box Design.” *Proceedings,Workshop in Selected Areas of Cryptography,SAC’96*,1996.
MITC90 Mitchell,C.;Walker,M.;and Rush,D. “CCITT/ISO Standards for Secure Message Handling.” *IEEE Journal on Selected Areas in Communications*,May 1989.
MITC92 Mitchell,C.;Piper,F.;and Wild,P. “Digital Signatures,” in [SIMM92a].
MIYA90 Miyaguchi,S.;Ohta,K.;and Iwata,M. “Confirmation that Some Hash Functions Are Not Collision Free.” *Proceedings,EUROCRYPT’90,1990*;published by Springer-Verlag.
MUFT89 Muftic,S.*Security Mechanisms for Computer Networks*.New York:Ellis Horwood,1989.
MURP90 Murphy,S. “The Cryptanalysis of FEAL-4 with 20 Chosen Plaintexts.” *Journal of Cryptology*,No.3,1990.
MURP95 Murphy,E.,Hayes,S. and Enders,M.TCP/IP:*Tutorial and Technical Overview*.Upper Saddle River:NJ:Prentice Hall,1995.
WYER91 Myers,L.*Spycomm:Covert Communication Techniques of the Underground*.Boulder,CO:Paladin Press,1991.
NACH97 Nachenberg,C. “Computer Virus-AntiVirus Corvolution.” *Communications of the ACM*,January 1997.
NECH92 Nechvatal,J. “Public Key Cryptography,” in [SIMM92a].
NEED78 Needham,R.,and Schroeder,M. “Using Encryption for Authentication in Large Networks of Computers.” *Communications of the ACM*,December 1978.
NEUH93 Neuhaus,S.*Statistical Properties of IDEA Session Keys in PGP*.Unpublished report,available from author neuhaus@informatik.uni-kl.de,13 June 1993.
NEUM90 Neumann,P. “Flawed Computer Chip Sold for Years.” *RISKSFORUM Digest*,Vol.10,No.54,October 18,1990.
NEUM93a Neuman,B.,and Stubblebine,S. “A Note on the Use of Timestamps as Nonces.” *Operating Systems Review*,April 1993.
NEUM93b Neuman,B. “Proxy-Based Authorization and Accounting for Distributed Systems.” *Proceedings of the 13th International Conference on Distributed Computing Systems*,May 1993.
NIST91 National Institute of Standards and Tecnology,*Glossary of Computer Security Terminology*.NISTIR4659,1991.
ODLY95 Odlizko,A. “The Future of Integer Factorization.” *CryptoByets*,Summer 1995.
OORS90 Oorschot,P.and Wiener,M. “Parallel Collision Search with Application to Hash Functions and Discrete Logarithms.” *Proceedings,Second ACM Conference on Computer and Communications Security*,1994.

OPPL97 Oppliger, R. “Internet Security: Firewalls and Beyond.” *communications of the ACM*, May 1997.
ORE67 Ore, O. *Invitation to Number Theory*. Washington, DC: The Mathmatical Association of America, 1967.
ORE76 Ore, O. *Number Theory and Its History*. New York: Dover 1976.
PARK88 Park, S., and Miller, K. “Random Number Generators: Good Ones are Hard to Find.” *Communications of the ACM*, Octobor 1988.
PFLE97 Pfleeger, C. *Security in Computing*. Upper Saddle River, NJ: Prentice Hall, 1997.
POHL81 Pohl, I., and Shaw, A. *The Nature of Computation: An Introduction to Computer Science*. Rockville, MD: Computer Science Press, 1981.
POPE79 Popek, G., and Kline, C. “Encryption and Secure Computer Networks.” *ACM Computing Surveys*, December 1979.
PORR92 Porras, P. *STAT: A State Transaction Analysis Tool for Intrusion Detection*. Master’s Thesis, University of California at Santa Barbara, July 1992.
PREN96 Preneel, B., and Oorschot, P. “0n the Security of Two MAC Algorithms.” *Proceedings, EOROCRYPT’96*, 1996; published by Springer-Verlag.
RABI78 Rabin, M. “Probabilistic Algorithms for Primality Testing.” *Journal of Number Theory*, December 1980.
RAND55 Rand Corporation. *A Million Random Digits*. New York: The Free Press, 1955.
RIBE96 Ribernboim, P. *The New Book of Prime Number Records*. New York: Springer-Verlag, 1996.
RIVE78 Rivest, R.; Shamir, A.; and Adleman, L. “A Method for Obtaining Digital Signatures and Public Key Cryptosystems.” *Communications of the ACM*, February 1978.
RIVE90 Rivest, R.; “The MD4 Message Digest Algorihm.” *Proceedings, Crypto’90*. August 1990; published by Springer-Verlag.
RIVE94 Rivest, R. “The RC5 Encryption Algorithm.” *Proceedings, Second* International *Workshop on Fast Software Encryption*, December 1994; publish“ed by Springer-Verlag.
RIVE95 Rvest, R. “The RC5 Encryption Algorithm.” *Dr.Dobb’s Journal*, January 1995.
RIVE97 Rivest, R. A *Description of the RC2(r) Encryption Algorithm*. Internet Draft draft □rivest-rc2desc-00.txtx, June 1997.
ROBS95 Robshaw, M. *Block Ciphers*. RSA Laboratories Technical Report TR-601, August 1995.
ROSE93 Rosen, K. *Elementary Number Theory and its Applications*. Reading, MA: Addison-Wesley, 1993.
RSA97 RSA Data Security, Inc. “Government Encryption Standard DES Takes a Fall.” *RSA Dta Press Relaese*, June 17, 1997. http://www.rsa.com/des.
RUBI97a Rubin, A.; Geer, D.; and Ranum, M. *Web Security Sourcebook*. New York: Wiley, 1997.
RUBI97b Rubin, A. “An Experience Teaching a Graduate Course in Cryptography.” *Cryptologia*, April 1997.

SAFF93 Safford ,D.;Schales,D.;and Hess,D. “The TAMU Security Package:An Ongoing Response to Internet Intruders in an Academic Environment.” *Proceedings, UNIX Security Symposium IV*, Octobor 1993.

SALO96 Salomaa,A. *Public-Key Cryptography*. New York:Springer-Verlag,1996.

SAUE81 Sauer,C.,and Chandy,K. *Computer Systems Performance Modeling*. Englewood Cliffs,NJ:Prentice Hall,1981.

SCHA96 Schaefer,E. “A Simplified Dta Encryption Standard Algorithm.” *Cryptologia*, January 1996.

SCHN91 Schnorr,C. “Efficient Signatures for Smart Card.” *Journal of Cryptology*, No.3,1991.

SCHN93 Schneier,B.Applied Cryptography.New York:Wiley,1996.

SEBE89 Seberry,J.,and Pieprzyk,J. *Cryptography:An Introduction to Computer Security*. Englewood Cliffs,NJ:Prentice Hall,1989.

SEME96 Semeria,C. *Internet Firewalls and Security*. 3Com Corp.,1996.Available at www.3com.com.

SHAN49 Shannon,C. “Communication Theory of Secrecy Systems.” *Bell Systems Technical Journal*, No.4.1949.

SIMM92a Simmons,G.,ed. *Contemporary Cryptology*:The Science of Information Integrity.Piscataway,NJ:IEEE Press,1992.

SIMM92b Simmons,G. “A Survey of information Authentication,” in [SIMM92a].

SIMM93 Simmonos,G. “Cryptology.” *Encyclopaedia Britannica, Fifteenth Edition*, 1993.

SINK66 sinkov,A. *Elementary Cryptanalysis*:A Mathematical Approach.Washington,DC:The Mathematical Assosoation of America,1966.

SMIT97 Smith,R. *Internet Cryptography*. Reading,MA:Addison-Wesley,1997.

SNAP91 Snapp,S.,et al. “A System for Distributed Intrusion Detection.” *Proceedings, COMPCON* Spring’91,1991.

SPAF92a Spafford,E. “Observing Reusable Password Choices.” *Proceedings, UNIX Security Symposium III*, September 1992.

SPAF92b Spafford,E. “OPUS:Preventing Weak Password Choices.” *Computers and Security,No.3,1992.*

STAL97 Stallings,W. *Data and Computer Communications,Fifth Edition*. Upper Saddle River,NJ:Prentice Hall,1997.

STEI88 Steiner,J.;Neuman,C;and Schiller,J. “Kerberos:An Authentication Service for Open Networked Systems.” *Proceedings of the Winter 1988 USENIX Conference*, February 1988.

STEP93 Stephenson,P. “Preventive Medicine.” *LAN Magazine*, November 1993.

STER92 Sterling,B. *The Hacker Crackdown:Law and Disorder on the Electronic Frontier*. New York:Bantam,1992.

STEV94 Stevens, *W.TCP/IP Illustrated,Volume 1:The Protocols*. Reading,MA:Addison-Wesley,1994.

STIN95 Stinson,D. *Cryptography:Theory and Practice*. Boca Raton,FL:CRC Press,1995.

STOL88 Stoll, C. "Stalking the Wily Hacker." *Communications of the ACM*, May 1988. Reprinted in [DENN90].
STOL89 Stoll, C. *The Cuckoo's Egg*. New York: Doubleday, 1989.
THOM84 Thompson, K. "Reflections on Trusting Trust (Deliberate Software Bugs)." *Communications of the ACM*, August 1984.
TIME90 Time, Inc. *Computer Security, Understanding Computer Series.* Alexandria, VA: Time-Life *Books, 1990.*
TIPP27 Tippett, L. *Random Sampling Numbers*. Cambridge, England: Cambridge University Press, 1927.
TSUD92 Tsudik, G. "Message Authentication with One-Way Functions." *Proceedings, INFOCOM'92*, May 1992.
TUCH79 Tuchman, W. "Hellman Presents No Shortcut Solutions to DES." *IEEE Spectrum*, July 1979.
VACC89 Vaccaro, H., and Liepins, G. "Detection of Anomalous Computer Session Activity." *Proceedings of the IEEE Symposium on Research in Security and Privacy,* May 1989.
VOYD83 Voydock, V., and Kent., S. "Security Mechanisms in High-Level Network Protocols." *Computing Surveys*, June 1983.
WAYN93 Wayner, P. "Should Encryption Be Regulated?" *Byte*, May 1993.
WAYN96 Wayner, P. *Disappearing Cryptography*. Boston: AP Professional Books, 1996.
WEBS86 Webster, A., and Tavares, S. "0n the Design of S-Boxes." *Proceedings, Crypto'85*, 1985; published by Springer-Verlag.
WEGE87 Wegenerm, I. *The Complexity of Boolean Functions*. New York: Wiley, 1987.
WIEN90 Wiener, M." Cryptanalysis of Short RSA Secret Exponents." *IEEE Transactions on Information Theory*, vol. IT-36, 1990.
WIEN93 Wiener, M. "Efficient DES Key Search." *Proceedings, Crypto'93*, 1993; published by Springer-Verlag.
WEIS93 Weiss, J., and Schremp, D. "Putting Data on a Diet." *IEEE Spectrum*, August 1993.
WILL76 Williamson, M. *Thoughts on Cheaper Non-Secret Encryption*. CESG Report, August 1976.
WOO92a Woo, T., and Lam, S. "Authentication' Revisited." *Computer*, April 1992.
YIN97 Yin, Y. "The RC5 Encryption Algorithm: Two Years On." *CryptoBytes*, Winter 1997.
YUVA79 Yuval, G. "How to Swindle Rabin." *Cryptologia*, July 1979.
ZENG91 Zeng. K.; Yang, C.; Wei, D.; and Rao. T. "Pseudorandom Bit Generators in Stream-Cipher Cryptography." *Computer*, February 1991.
ZIV77 Ziv, J., and Lempel, A. "A Universal Algorithm for Sequential Data Compression." *IEEE Transactions on Information Theory*, May 1977.

색 인

ㄱ

ㄴ

ㄷ

ㄹ

ㅁ

ㅂ

ㅅ

ㅈ

ㅊ

ㅋ

ㅌ

ㅍ

ㅎ

ㅆ

ㅇ

* * *